A Framework for Vision-based Static Hand Gesture Recognition

Dipak Kumar Ghosh

Department of Electronics and Communication Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India

A Framework for Vision-based Static Hand Gesture Recognition

Thesis submitted in partial fulfillment of the requirements for award of the degree

DOCTOR OF PHILOSOPHY

by

Dipak Kumar Ghosh

Under the supervision of

Dr. Samit Ari

Department of Electronics and Communication Engineering

National Institute of Technology Rourkela

Rourkela-769008, INDIA

September 2016

Department of Electronics and Communication Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India

Supervisor's Certificate

This is to certify that the work presented in this thesis entitled **A Framework for Vision-based Static Hand Gesture Recognition** by **Dipak Kumar Ghosh**, Roll Number: 510EC106 submitted to the National Institute of Technology Rourkela for the degree of **Doctor of Philosophy**, is a record of original research work, carried out by him in the Dept. of Electronics and Communication Engineering under my supervision and guidance. I believe that the thesis fulfills part of the requirements for the award of degree of **Doctor of Philosophy** in Electronics and Communication Engineering. To the best of my knowledge, neither this thesis nor any part of it has been submitted for any degree or diploma to any other University/Institute in India or abroad.

Place: NIT, Rourkela
Date: September 27, 2016

Dr. Samit Ari
Assistant Professor
Department of Electronics and Communication Engineering,
National Institute of Technology,
Rourkela, Odisha,
INDIA 769 008.

Dedicated to

the sacrifice and endurance of my parents

Declaration of Originality

I, Dipak Kumar Ghosh, Roll Number: 510EC106 hereby declare that the thesis entitled **A Framework for Vision-based Static Hand Gesture Recognition** is a bonafide record of my original research work carried out as a doctoral student of NIT Rourkela under the guidance of Dr. Samit Ari and, to the best of my knowledge, it contains no material previously published or written by another person, nor any material presented for the award of any other degree or diploma of NIT Rourkela or any other institution. Any contribution made to this research by others, with whom I have worked at NIT Rourkela or elsewhere, is explicitly acknowledged in the thesis. Works of other authors cited in this thesis have been duly acknowledged under the section "References". I have also submitted my original research records to the scrutiny committee for evaluation of my thesis.

I am fully aware that in case of any non-compliance detected in future, the Senate of NIT Rourkela may withdraw the degree awarded to me on the basis of the present dissertation.

Place: NIT, Rourkela
Date: September 27, 2016

Dipak Kumar Ghosh

Acknowledgment

Throughout my PhD work I came across many people whose support helped me to complete this research work and at this moment I would like to take the opportunity to acknowledge them. First and foremost I would like to express my deep and sincere gratitude towards my respectable supervisor, Prof. Samit Ari for his invaluable guidance, constant inspiration and motivation along with enormous moral support during my difficult phase to complete this work, without his suggestions and ideas, this thesis would not be an asset for me. I am indebted to him for the valuable time he has spared for me during this work.

I am very much thankful to Prof. K. K. Mahapatra, Professor & Head of the Department, Electronics and Communication Engineering, for his continuous encouragement. Also, I am indebted to him for providing me with all the official and laboratory facilities.

I am also thankful to former Director, Prof. S. K. Sarangi and Director, Prof. R. K. Sahoo, National Institute of Technology, Rourkela for allowing me to avail the necessary facilities of the Institute for completion of this work. I am grateful to my DSC members Prof. D. P. Acharya, Prof. S. K. Jena and Prof. D. Patra, for their valuable comments and suggestions.

I would like to thank Prof. S. K. Patra, Prof. S. K. Behera, Prof. A. K. Swain, Prof. S. K. Das, Prof. A. K. Sahoo, Prof. L. P. Ray, Prof. U. K. Sahoo, Prof. S. Sarkar and Prof. S. K. Kar whose encouragement helped me to work hard. They have been great sources of inspiration for me and I thank them from bottom of my heart. I thank to all the non-teaching staff and technical staff of Dept. of ECE, especially Mr. B. Das and

Mr. Kujur for their support.

I acknowledge all research scholars, friends and juniors of Dept. of Electronics and Communication Engineering, NIT, Rourkela for their generous help in various ways to complete the thesis work. I should also thank my friend Manab Kumar Das with whom I shared many of the ideas related to research work and who gave me invaluable feedback.

I would like to acknowledge my parents, sister and brothers for their support, strength and motivation. A special thank goes to my father, Manik Chandra Ghosh and mother, Karuna Ghosh for their love, patience and understanding throughout these years. I have realized that without the selfless help from them, I could never achieve this goal. I would like to convey my heartiest regards to my parents for their boundless love and affection.

Finally, my greatest regards to the Almighty for giving me the strength and patience to complete the doctoral research work through all these years.

Dipak Kumar Ghosh

Roll Number: 510EC106

Place: NIT, Rourkela

Date: September 27, 2016

Abstract

In today's technical world, the intellectual computing of a efficient human-computer interaction (HCI) or human alternative and augmentative communication (HAAC) is essential in our lives. Hand gesture recognition is one of the most important techniques that can be used to build up a gesture based interface system for HCI or HAAC application. Therefore, suitable development of gesture recognition method is necessary to design advance hand gesture recognition system for successful applications like robotics, assistive systems, sign language communication, virtual reality etc. However, the variation of illumination, rotation, position and size of gesture images, efficient feature representation, and classification are the main challenges towards the development of a real time gesture recognition system. The aim of this work is to develop a framework for vision based static hand gesture recognition which overcomes the challenges of illumination, rotation, size and position variation of the gesture images. In general, a framework for gesture recognition system which consists of preprocessing, feature extraction, feature selection, and classification stages is developed in this thesis work. The preprocessing stage involves the following sub-stages: image enhancement which enhances the image by compensating illumination variation; segmentation, which segments hand region from its background image and transforms it into binary silhouette; image rotation that makes the segmented gesture as rotation invariant; filtering that effectively removes background noise and object noise from binary image and provides a well defined segmented hand gesture. This work proposes an image rotation technique by coinciding the first principal component of the segmented hand gesture with vertical axes to make it as rotation invariant. In the feature extraction stage, this work extracts

localized contour sequence (LCS) and block based features, and proposes a combined feature set by appending LCS features with block-based features to represent static hand gesture images. A discrete wavelets transform (DWT) and Fisher ratio (F-ratio) based feature set is also proposed for better representation of static hand gesture image. To extract this feature set, DWT is applied on resized and enhanced grayscale image and then the important DWT coefficient matrices are selected as features using proposed F-ratio based coefficient matrices selection technique. In sequel, a modified radial basis function neural network (RBF-NN) classifier based on k-mean and least mean square (LMS) algorithms is proposed in this work. In the proposed RBF-NN classifier, the centers are automatically selected using k-means algorithm and estimated weight matrix is updated utilizing LMS algorithm for better recognition of hand gesture images. A sigmoidal activation function based RBF-NN classifier is also proposed here for further improvement of recognition performance. The activation function of the proposed RBF-NN classifier is formed using a set of composite sigmoidal functions. Finally, the extracted features are applied as input to the classifier to recognize the class of static hand gesture images. Subsequently, a feature vector optimization technique based on genetic algorithm (GA) is also proposed to remove the redundant and irrelevant features. The proposed algorithms are tested on three static hand gesture databases which include grayscale images with uniform background (Database I and Database II) and color images with non-uniform background (Database III). Database I is a repository database which consists of hand gesture images of 25 Danish/international sign language (D/ISL) hand alphabets. Database II and III are indigenously developed using VGA Logitech Webcam (C120) with 24 American Sign Language (ASL) hand alphabets.

Keywords: American Sign Language (ASL) hand alphabet, combined features, discrete wavelet transform (DWT), Danish/international sign language (D/ISL) hand alphabet, Fisher Ratio (F-ratio), gesture recognition, genetic algorithm (GA), k-means algorithm; least-mean-square algorithm (LMS), localized contour sequences (LCS), multilayer perceptron back propagation neural network (MLP-BP-NN), radial basis function neural network (RBF-NN), static hand gesture.

Contents

List of Abbreviations

ASL	American sign language
ANNs	Artificial neural networks
ANFIS	Adaptive neuro-fuzzy inference system
BLOB	Binary Large Object-analysis
CSL	Chinese sign language
CWT	Continuous wavelet transform
CSRBF	Composite sigmoidal radial basis function
CM	Complex moments
CM-MLP-BP-NN	Complex moments with MLP-BP-NN classifier
DWT	Discrete wavelet transform
D/ISL	Danish/international sign language
DCT	Discrete cosine transformation
DWT-FR	DWT and F-ratio based features
DWT-FR-GA-KLMS-RBFNN-GAF	GA based optimized DWT-FR features with KLMS-RBFNN-GAF classifier
DWT-FR-GA-KLMS-RBFNN-SAF	GA based optimized DWT-FR features with KLMS-RBFNN-SAF classifier
DCT-KNN	Discrete cosine transform coefficients with k-nearest neighbors classifier
F-ratio	Fisher ratio
FD	Fourier descriptor

FCM	Fuzzy C-mean
FS	Feature selection
FSS	Feature subset selection
FN	False negative
FP	False positive
FPR	False positive rate
FT	Fourier transform
GA	Genetic algorithm
HCI	Human-computer interaction
HAAC	Human alternative and augmentative communication
HMI	Human Machine Interface
HU-MLP-BP-NN	Hu moment invariants with MLP-BP-NN classifier
IDCT	Inverse discrete cosine transform
KNN	K-nearest neighbor
KM	Krawtchouk moments
KLMS-RBFNN-GAF	K-mean and LMS based RBF neural network with Gaussian activation function
KLMS-RBFNN-SAF	K-mean and LMS based RBF neural network with composite sigmoidal activation function
KM-MD	Krawtchouk moments with minimum distant classifier
LCS	Localized contour sequences
LMS	Least mean square
LLE	Local linear embedding
MLP	Multilayer perceptron
MLP-BP-NN	Multilayer perceptron back propagation neural network
MD	Minimum distance

MSE	Mean square error
MSE_{train}	Mean square error in training phase
NSDS-KLAD	Normalized silhouette distance signal features with k-least absolute deviations classifier
PCA	Principal component analysis
RBF	Radial basis function
RBF-NN	Radial basis function neural network
ROC	Receiver operating characteristic
RMS	Root mean square
SVM	Support vector machine
STFT	Short time Fourier transform
TN	True negative
TP	True positive
TPR	True positive rate
WT	Wavelet transform

List of Symbols

$\psi_{m,n}(x)$	Mother wavelet function or wavelet function
$\phi_{m,n}(x)$	Scaling function
G_0	High-pass filter
H_0	Low-pass filter
$g(k)$	Impulse response of filter G_0
$h(k)$	Impulse response of filter H_0
$A1$	Approximation coefficients at decomposition level 1
$D1^H$	Horizontal detail coefficients at decomposition level 1
$D1^V$	Vertical detail coefficients at decomposition level 1
$D1^D$	Diagonal detail coefficients at decomposition level 1
$K_n(x;p,N)$	The classical Krawtchouk polynomial of order n
${}_2F_1$	The hyper geometric function
$(a)_m$	The Pochammer symbol
$\delta[.]$	The Kronecker delta function
$\mathbf{x}$	An input vector
$\mathbf{t}$	Target vector
$G_i(\mathbf{x})$	Gaussian function
σ	Spread factor or width of Gaussian function
∇	Gradient operator
η	Learning rate parameter

p_c	The crossover probability
p_s	The swapping probability
p_m	The mutation probability
Acc	Recognition accuracy
Sen	Recognition sensitivity
Spe	Recognition specificity
Ppr	Recognition positive predicitivity

List of Figures

List of Tables

CHAPTER 1

Introduction

In the present scenario of intelligent computing, efficient human computer interaction (HCI) or human alternative and augmentative communication (HAAC) is becoming extremely important in our daily lives [1,2]. Gesture recognition is one of the important research approaches in HCI or HAAC applications. In general, gestures are communicative, meaningful body motions or body language expressions involving physical actions of the fingers, hands, arms, face, head or body with the intent of conveying meaningful information or interacting with the environment [1]. Since hands are more flexible and controllable parts of human body, they are used to perform most of the important body language expressions [3, 4]. Therefore, hand gestures are suitable for exchanging information such as representing a number, pointing out an object, expressing a feeling etc. Hand gestures are also used as a primary interaction tool for sign language and gesture based computer control [4]. In a very common HCI system, simple mechanical devices like keyboard and mouse are used for man-machine interaction. However, these devices inherently limit the speed and spontaneity of the interaction between man and the machine [5]. On the other hand, in recent years, the interaction methods using hand gesture based computer vision have become a popular alternative communication modalities for man-machine interaction due to its natural interaction ability [6, 7]. A suitable design of hand gesture recognition framework can be used to develop an advanced hand gesture based interface system for successful HCI or HAAC applications like robotics, sign language communication, virtual reality etc [8, 9].

1.1 Hand Gestures

Hand gestures are a collection of movements or a pose or configurations of hand and arm performed by one or two hands, used to signify a certain meaning or used to communicate with others. Since the ability of the human hand to acquire a huge number of clearly observable configurations, hand gestures become the largest category of gestures among all gesture classes. Based on different application scenarios, hand gesture can be categorized into several classes like conversational gestures, communicative gestures,

controlling gestures and manipulative gestures [10]. In general, people often make hand movements that are synchronized with their speech. Traditionally, these movements are called conversational hand gestures which convey semantic information in addition with speech [11]. Sign language is an important example of communicative gestures. Sign language is highly structural and it can help a deaf one to interact with ordinary people or computers [12]. Controlling gestures are used in remote control applications such as consumer electronic control system [13, 14] and robot control [15]. Manipulative gestures assist as a natural way to interact with virtual objects. Teleoperation and virtual assembly are good examples for the applications of manipulative gestures [10]. In general, gestures can be classified into static gestures or postures [5, 13, 16–18] and dynamic gestures [19–22]. A static gesture is described in the form of definite hand configuration or poses while dynamic gesture is a moving gesture, articulated as a sequence of hand movements and arrangements [9]. Static gesture does not carry the time-varying information. Therefore, it can be completely analyzed using a single image or a set of images of the hand taken at a specific time. A good example of static hand gesture is an image of hand sign such as the "OK", or the "STOP", which is enough for complete understanding of the meaning. On the other hand, dynamic gesture is a video sequence of hand movements. Hence, a sequence of hand images connected by motion over a short time span is required to analyze a dynamic gesture. The simple example of a dynamic gesture is "goodbye" or "come here" and it can only be recognized by taking the temporal context information [7]. Since static gestures can convey certain meanings and sometimes act as specific transition states of dynamic gestures, recognition of static gesture is one of the most important parts in the area of gesture recognition [2, 23].

1.2 Hand Gesture Recognition

Hand gesture recognition is a process to recognize the gestures performed by human hand or hands. The research objective of hand gesture recognition is to build a system framework which can recognize hand gestures that convey certain informa-

tion. According to sensing techniques, hand gesture recognition methods are broadly classified into two categories: (i) glove-based technique [24–27] and (ii) vision-based technique [5, 9, 13, 16, 28, 29].

In glove-based techniques, sensors (mechanical or optical) attached to a glove are utilized to measure the joint angles, positions of the fingers and position of the hand for determining the hand gesture [24]. In [25], the authors have implemented a tool for directing robotic actions where a Cyberglove with Polhemus sensor is successfully used to transform operator hand motions into virtual robot end-effector motions as a tool for directing robotic actions. Parvini *et al.* [26] have developed a hand gesture recognition system by utilizing bio-mechanical characteristics of hand where CyberGlove is used as a virtual reality user interface to acquire data corresponding to 22 static hand gesture of American sign language (ASL) hand alphabets. In [27], the authors have designed a glove-based hand gesture recognition system to classify 24 static hand gesture of ASL alphabet where 5DT Data Glove 5 Ultra is used to capture information of the flexion degree of each finger of the hand and a 3-axis accelerometer placed on the back of the hand to know its orientation. However, gloves are quite expensive and the weight of the gloves and associated measuring equipment restrict the free movement of the hand. Therefore, the user interface is complicated and less natural for glove-based techniques [30].

On the other hand, vision-based techniques use one or more cameras to capture the gesture images and provide more natural and non-contact solutions for HCI [28, 31]. In [13], the authors have developed a system where a webcam is used to capture the hand gestures and then registration, normalization, and feature extraction process are applied for ultimate classification and to operate the remote controller. Gupta and Ma [16] have developed an automatic hand gesture classification system based on gesture contours where a single video camera and a real-time video software are used for the real-time acquisition, sampling, quantizing, and storing of the gesture images. In [5], authors have developed a static hand gesture recognition system where the hand gesture images are captured using an RGB Frontech e-cam which is connected to an Intel

core-II duo 2 GB RAM processor. The vision-based technique becomes more popular because of the following advantages: (i) it does not demand any wearable or attached electromechanical equipment, i.e. it is a passive interface; (ii) it only uses inexpensive one or more cameras for image acquisition; (iii) this technique is quite simple and provides a natural interaction between humans and computers [30].

1.3 Applications of Hand Gesture based Interface System

Hand gestures based interface system can be used in various applications [32]. Hand gesture can facilitate deaf and dumb people to communicate with ordinary people or computer. The natural HCI for virtual environments can be achieved using hand gestures. A few application areas of hand gesture based interface system are follows.

1.3.1 Robotics and Telepresence

The demand of manual operations may raise in some cases like system failure or inaccessible remote areas or emergency hostile conditions. In many cases, it is impossible for human operators to present near the machines [33]. Robotics and telepresence using hand gestures have a great use in this situation. The aim of telepresence is to provide the physical operational support to accomplish the necessary task by mapping the operator arm to the robotic arm. The researchers of the University of California, San Diego have designed real time ROBOGEST system to control an outdoor autonomous vehicle in a natural way using hand gestures [34]. Hand gesture recognition system can used to control the robot operations as reported in [35, 36] where a hand gesture recognition system is connected with robot to control different operations based on the given hand gestures.

1.3.2 Virtual Reality

Virtual reality is a computer-simulated environment which is presented to the user in such a way that the user accepts it as a real environment. Virtual reality environments

are primarily sensory experiences which are displayed either on a computer screen or with special stereoscopic displays [37]. The use of gestures in virtual reality applications has become the greatest inspiration in computing [38]. In virtual reality, gestures are used to enable realistic manipulations of virtual objects for 3D display interactions [39].

1.3.3 Vehicle Interfaces

The primary motivation of hand gesture based vehicle interfaces is the use of hand gestures for secondary controls. In [40], the authors have described an approach of vision-based gesture recognition for human-vehicle interactions such as calling-up vehicle, stopping vehicle, and directing vehicle etc. In [41], Pickering *et al.* has described the primary and secondary driving task, the trends and issues of Human Machine Interface (HMI) for driving automotive user interface where hand gesture recognition is considered as a realistic substitute for user controls.

1.3.4 Sign Language

Sign language is one of the important examples of communicative gestures. It is used to help deaf and dumb people to communicate with normal people or computer [12]. A sign language recognition system could allow a deaf to communicate with other people without any interpreter and sometimes it could be used to generate speech or text corresponding to different sign languages that makes the deaf or dumb more independent [33]. At the same time, the sign language recognition system has become one of the most important systems for HCI applications [42, 43].

1.3.5 Desktop and Tablet PC Applications

An alternative to mouse and keyboard, the gestures based interaction is used in desktop computing applications [44]. Many gestures such as pen-based gestures are used for desktop computing tasks which involve manipulating graphics, or annotating and editing documents [45]. Recently, a gesture recognition technology is introduced by

eyeSight (a company expert in Machine Vision and gesture recognition technology) for operating Android Tablets and Windows-based portable computers [28].

1.3.6 Healthcare

The hand gesture recognition system offers a possible alternative to reduce infection rates at hospitals. The touch screen system can be replaced by gesture based system in many hospital operating rooms which must be sealed to prevent accumulation of dust or spreading of contaminants [28]. Wachs *et al.* [46] have developed a hand gesture recognition system where doctors use hand gestures instead of touch screens or computer keyboards for navigation and manipulation of digital images during medical processes.

1.3.7 Consumer electronic control

Hand gesture recognition system can be used to control consumer electronics like television [13]. In this application, a gesture recognition system is used to generate the commands like '$TV1'$, '$Video1'$, '$Volume\,Up'$, '$Volume\,Down'$, '$Channel\,Up'$, '$Channel\,Down'$, '$Power'$, 'TV/AV', '$Play'$ and '$Stop'$, for controlling different television operations as per given hand gestures.

1.4 A Brief Overview of Vision-based Hand Gesture Recognition System

The aim of the vision-based hand gesture recognition system is to process, analyze and recognize the hand gesture image. The overall process of vision-based hand gesture recognition system can be divided into two phase: (a) enrollment or training phase and (b) recognition or testing phase. The schematic diagram of a vision-based hand gesture recognition system is shown in Figure 1.1. Both training and testing phases include the same preprocessing, feature extraction and feature vector optimization steps. In general, a vision-based static hand gesture recognition system consists of the following stages: preprocessing, feature extraction, feature vector optimization and classification. The

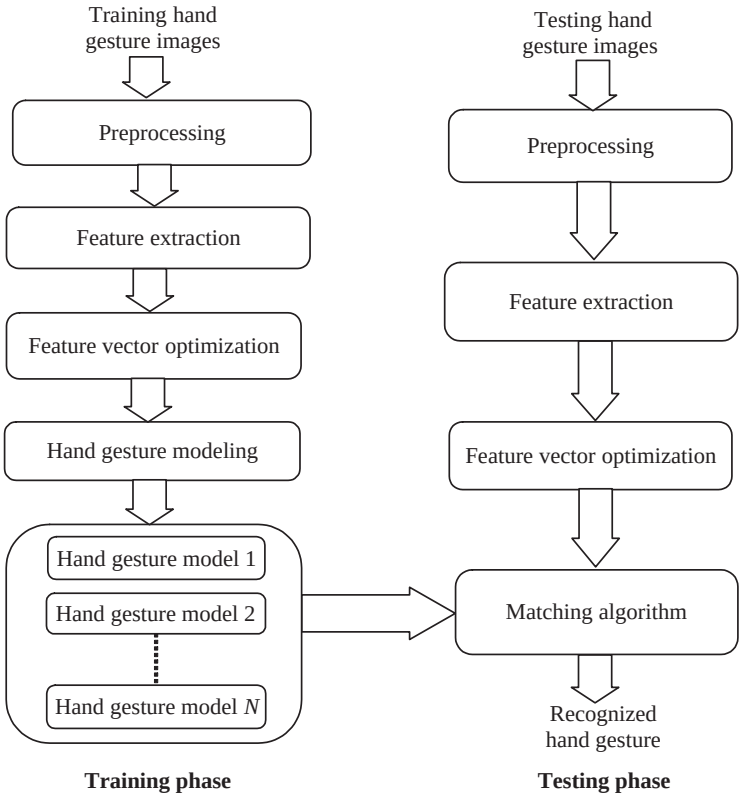


Figure 1.1: The schematic diagram of a vision-based hand gesture recognition system.

preprocessing stage includes following sub-stages: (i) Image enhancement: enhances the image by compensating illumination variation; (ii) Segmentation: segments hand region from its background image and transforms into binary silhouette; (iii) Image rotation: makes segmented gesture rotation invariant and (iv) Filtering: effectively removes background noise and object noise from the binary image and provides a well defined segmented hand gesture [9]. The extraction of efficient features is an important stage in a hand gesture recognition system. The aim of feature extraction is to extract the most discriminative and relevant characteristics of hand gesture image for recognition. In this way, the hand gesture image is represented by a set of feature or features in a compressed manner. Various features have been reported in the literature to represent static hand gesture including features like statistical moments [47], block-

based feature [17], Haar-link feature [48], Fourier descriptors (FD) [18], vector lengths between the gesture's centroid to the useful portion of the gesture border [12], localized contour sequence (LCS) [16] and normalized silhouette distance signal features [49] etc. In feature vector optimization stage, the extracted feature vectors are further optimized by removing redundant and irrelevant features. The aim of the feature vector optimization technique is to reduce the dimension of feature vector. This minimizes the feature space complexity keeping the performance of hand gesture recognition reasonably high. It involves definition of most discriminative and informative features of the original data for recognition. This can be performed by eliminating the unwanted and redundant features [50]. The advantages of feature vector optimization technique for a hand gesture recognition system are as follows: (i) induction of faster classification in final mode due to the reduction of feature vector dimension and (ii) achievement of reasonably high recognition performance due to the elimination of unwanted and redundant features [50]. The optimized feature set is applied to the input of the classifier for recognition of hand gesture image. At the training phase, optimized feature sets are used to train the classifier and trained gesture models are stored at the end of this phase. The collection of trained hand gesture models is called a hand gesture model database. At the time of recognition or testing phase, the class of an unknown hand gesture image is determined by comparing it with the models of known hand gesture images. The best matching indicates the class of unknown hand gesture image. A variety of methods [12, 13, 16, 17, 51, 52] have been reported to classify hand gestures like multilayer perceptron back propagation neural network (MLP-BP-NN) [13], minimum distance (MD) classifier [16], fuzzy C-mean (FCM) classifier [17], k-nearest neighbor (KNN) [51] classifier etc.

Over the last few decades, a number of researchers have reported a variety of static hand gesture recognition techniques [5, 7, 12, 13, 16–18, 51, 53, 54]. Pavlovic *et al.* [55] have surveyed the literature on visual interpretation of hand gestures in the context of its role in HCI. They have reported the fundamental methods which are used for modeling, analyzing, and recognizing the gestures and applications of gesture-based systems. Al-

Jarrah *et al.* [12] have presented a system for automatic translation of gestures of the manual alphabets in the Arabic sign language. The system is accomplished by training a set of adaptive neuro-fuzzy inference system (ANFIS) models, each of which is dedicated for the recognition of a given gesture. In feature extraction scheme, 30 vectors are computed to represent the hand gesture image. The system achieved a recognition rate of 93.55% using approximately 9 rules per ANFIS model. However, the performance of this reported technique depends on illumination variation of the images. In [16], Gupta and Ma have developed a complete system capable of robustly classifying hand gestures for HCI and HAAC applications using localized contour sequences (LCS). The system recognizes 10 hand gestures (0 to 9) of ASL using the best matching with a linear or nonlinear alignment method. The LCS representation is robust to contour noise, however, it is sensitive to the starting point of the contour sequences. In [52], the authors have developed a system for recognition of static hand gesture images based on the matching of boundary pixels using Hausdorff distance. They have reported that the system is achieved an average recognition rate of 90% for 26 hand postures. However, the matching technique based on Hausdorff distance is computationally complex. Conceil *et al.* [53] have reported a vision-based approach for hand posture recognition using Fourier descriptor (FD). For their own database, the 11 classes of hand posture have been recognized with a recognition rate of 84.58% using a Bayesian distance classifier. Bourennane and Fossati [18] have compared several shape descriptors for hand posture recognition. They have extracted two sets of Fourier descriptors (FD1 and FD2) and compared them with Hu invariants [56] and Zernike moments [57]. The hand postures are classified using Euclidean distance classifier. The experiments were conducted on Triesch database [58] with uniform background images and their own developed database of 11 hand gestures. The results show that FD1 provides higher recognition rate compared to FD2, Hu invariants and Zernike moments. Chenglong *et al.* [54] have reported a new technique based on a combinational feature vector and multi-layer perceptron (MLP) for hand gesture recognition. The combinational feature vector is formed using feature parameters of Hu invariant moment, hand gesture region, and Fourier descriptor. The

system achieves 97.4% recognition rate for 14 classes of hand gesture images. The contour of hand gesture is efficiently represented by Fourier descriptors (FDs). However, the recognition performance is sensitive to the variation of starting point and the number of FDs.

In [43], Teng *et al.* have reported a hand gesture recognition system based on local linear embedding (LLE). In feature extraction process, the high dimensional data is mapped to a low dimensional space using LLE where each point is approximated by a linear combination of their neighbors. The reported approach archives an average recognition rate of 90% for 30 hand gestures of Chinese sign language (CSL) alphabet. The LLE approach is invariant to translation and scale, however, it is sensitive to rotation. In [59], Hanning *et al.* suggested an approach to recognize the static hand gesture using local orientation histogram features. The summarized information of the hand image orientations are represented by the orientation histograms. The local orientation histogram features are illumination invariant, however, they are not rotation invariant. Triesch and von der Malsburg [58] have developed a technique for classification of hand postures against uniform and complex backgrounds. The reported technique uses elastic graph matching to classify hand postures and achieved a recognition rate of 92.9% and 85.8% for uniform and complex background images respectively. The matching procedure of the reported technique is user independent and robust in complex backgrounds. However, it is a complex procedure and sensitive to the variations of the viewpoint and hand anatomy. Wachs *et al.* [17] have suggested a methodology of cluster labeling and parameter estimation for the automated setup of a hand-gesture recognition system. In this reported technique, the hand gesture images are recognized by the system using block-based features and fuzzy C-means (FCM) classifier. The reported technique recognizes hand gesture with 98.9% and 93.75% for their own database of 13 gestures and Gripsee database [35] respectively. In [42], Munib *et al.* have developed a recognition technique to recognize the static gestures of ASL based on Hough transform and neural networks. The reported technique achieved a recognition rate of 92.3% for 20 classes of selected ASL signs. Shanableh and Assaleh [51] have presented a solution for recognition

of isolated Arabic sign language gestures. They have used the video-based gestures to segment out the hands of the signer based on color segmentation of the colored gloves. They have encapsulated the movements of the segmented hands in a bounding box. A discrete cosine transformation (DCT)-based feature set is extracted from the encapsulated images of the projected motion DCT and Zonal coding to the DCT coefficients with varying the cutoff values. The feature extraction schemes are validated using k-nearest neighbors (KNN) and polynomial classifier. An average recognition rate of 87% is achieved for KNN classifier. Huang *et al.* [60] have presented a technique to recognize 11 classes of hand gesture image using Gabor filters and support vector machine (SVM) classifiers. The reported technique achieved a recognition rate of 96.1% using Gabor-PCA features and SVM classifier. It uses Gabor filters for both stages, rotation estimation and feature extraction. However, Gabor filter based rotation estimation and feature extraction techniques are computationally complex.

Premaratne and Nguyen [13] jointly have developed a consumer electronics control system based on hand gesture moment invariant features and MLP-BP-NN classifier. They used the skin color segmentation to segment control gesture and normalized it using morphological filtering approach. They have shown that for limited number of hand gestures, 100% accuracy is achieved to control consumer electronics by their developed system. However, this system may fail to recognize hand gestures having similar shape. Priyal and Bora [5, 47] have reported the use of geometric and orthogonal moment features for hand gesture recognition. They have compared the performance of Krawtchouk, Geometric, Zernike and Tchebichef moments feature sets using minimum distant (MD) classifier to recognize 10 static hand gestures. They concluded that the Krawtchouk moments (KM) based feature set is more robust compared to Geometric, Zernike and Tchebichef moment based feature sets for hand gesture recognition. Hasan and Kareem [7] have presented a hand gesture recognition technique using shape analysis and MLP-BP-NN classifier. The reported technique recognizes a set of six specific static hand gestures including Open, Close, Cut, Paste, Maximize, and Minimize using two separate feature sets: hand contours and complex moments. It has been shown that

the technique achieves a recognition rate of 70.83% and 86.38% for hand contour and complex moment features respectively. However, moment based features have failed to recognize hand gestures having the similar shapes.

From the above study, it is seen that most of the reported techniques are applied to recognize hand gesture for limited number of gesture classes. It is also seen that there are several challenges including illumination variation, changes of position, size and rotation angle of the hand gesture images to recognize hand gesture images. Therefore, the development of a vision-based hand gesture recognition technique which will be able to detect a large class of hand gestures under the variations of illumination, rotation, position and size of the gesture images, is still a challenging task.

1.5 Contribution in the Thesis

This thesis deals with static hand gesture images of sign language and allows to recognize gesture even in a variation of illumination, rotation, position and size of gesture images. The major contributions of the thesis can be summarized as follows:

- Proposition of a novel technique to make hand gesture image as rotation invariant and a combined feature set for proper representation of static hand gesture image. In the proposed technique, the rotation normalization is achieved by coinciding the first principal component of the segmented hand gestures with vertical axes. In the feature extraction stage, two types of features are extracted from preprocessed hand gesture image: (i) Localized contour sequences (LCS): carry the contour information of the hand gesture images and (ii) Block-based features: incorporate the regional information of the hand gesture image. A combined feature set which couples LCS features with block-based features is also proposed in this work for better representation of static hand gesture image.

- Proposition of a discrete wavelet transform (DWT) and F-ratio based feature set to represent static hand gesture images. To extract this feature set, DWT is applied on resized and enhanced grayscale image and then the important DWT

coefficient matrices are selected as features using the proposed Fisher ratio (F-ratio) based technique. The proposed DWT and F-ratio based feature set carry localized information of hand gesture image in time and spatial frequency domains which can improve the hand gesture recognition performance.

- Proposition of modified radial basis function (RBF) neural network classifier for recognition of hand gesture images. This work proposes two modified RBF neural network (RBF-NN) classifiers: (i) K-mean and least-mean-square (LMS) based RBF-NN with Gaussian activation function (represented as KLMS-RBFNN-GAF classifier) and (ii) K-mean and LMS based RBF-NN with composite sigmoidal activation function (represented as KLMS-RBFNN-SAF classifier) where activation function is formed based on a set of composite sigmoidal functions. For both proposed classifiers, the centers are automatically selected using k-means algorithm and estimated weight matrix is further updated using LMS algorithm at the training phase to improve the recognition performance. The selected centers and updated weight matrix are stored at the end of training phase and used to recognize the unknown hand gesture images during testing phase.

- Proposition of a Genetic Algorithm (GA) based feature vector optimization technique to select an optimal subset of features from original set for hand gesture recognition. The aim of this technique is to reduce the dimension of feature vector which in turn helps to reduce the computational complexity of the recognition system while an attempt is made simultaneously to keep the performance accuracy reasonably high for hand gesture recognition. In this work, GA is applied to reduce the dimension of the extracted feature set by eliminating redundant and irrelevant features. Finally, the optimized feature subsets are used as input to the proposed classifier for hand gesture recognition.

1.6 Description of the Hand Gesture Databases

Three different hand gesture image databases which are mentioned as Database I, Database II and Database III, are used in this thesis. Databases I and II include grayscale hand gesture images with uniform background whereas Database III consists of color hand gesture images with nonuniform background. Database I is a repository database of 25 Danish/international sign language (D/ISL) hand alphabets which is publically available [61]. One static gesture image for each of 25 D/ISL hand alphabets are shown in Figure 1.2 [2]. This database consists of 1000 grayscale image for 25 static hand gestures, 40 samples are taken for each class with spatial resolution 256×248.

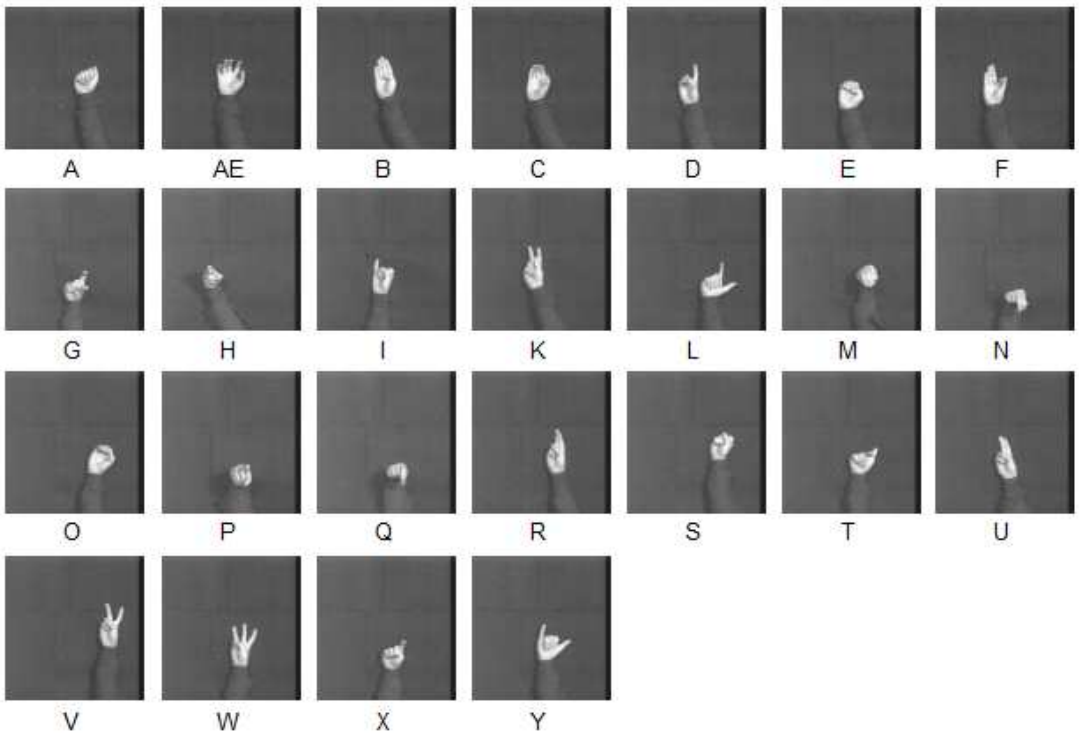


Figure 1.2: One static gesture image for each of 25 D/ISL hand alphabets.

Databases II and III are indigenously developed. In these databases, static gesture image of 24 ASL hand alphabets is captured using VGA Logitech Webcam (C120) [9]. One static gesture image for each of 24 ASL hand alphabets with uniform background and nonuniform background are shown in Figure 1.3 and Figure 1.4 respectively. Both

the databases are constructed by capturing ASL gesture images from 10 volunteers, most of them are not familiar with gesture recognition. Each volunteer provides a guide of postures as appear in the ASL browser developed by Michigan State University [62] and BGU-ASL DB [17].

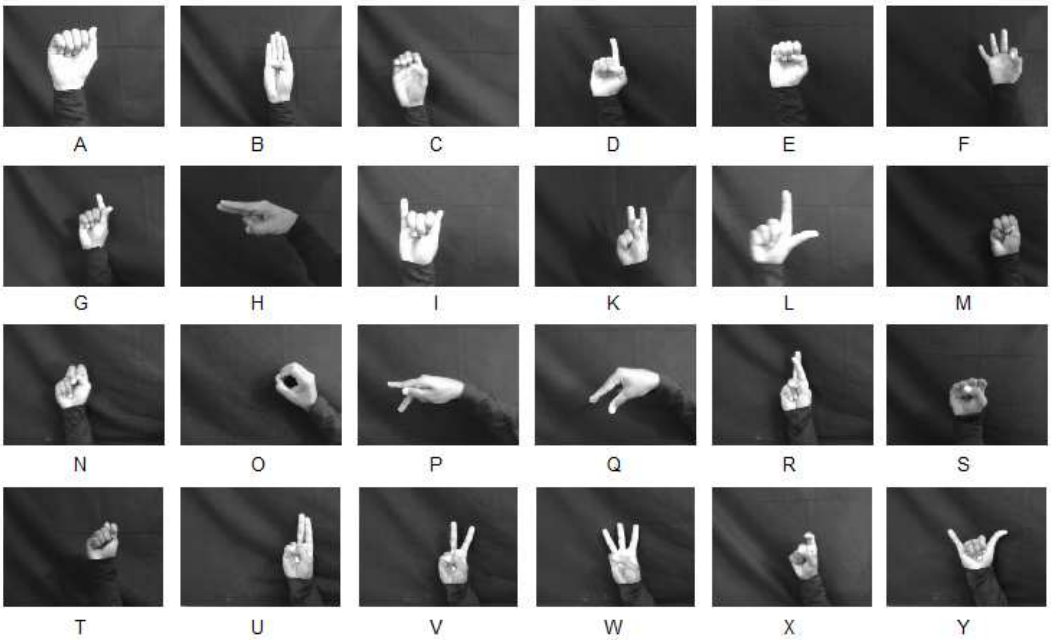


Figure 1.3: One static gesture image for each of 24 ASL hand alphabets with uniform background.

Database II comprises of 2400 grayscale images of 24 ASL hand alphabets with spatial resolution 320×240. Each ASL hand alphabet contains 100 images with 10 samples per volunteer. Each ASL hand alphabets are performed by every volunteer against an uniform black background based on different angles, positions and distances from the camera under different lighting condition. Database III also contains 2400 color images of 24 ASL hand alphabets with spatial resolution 320×240. Each ASL hand alphabet comprises of 100 images; 10 samples are performed by each volunteer against a nonuniform background with variations of angle, position and size changes under different lighting condition. This nonuniform background allows to test the robustness of the gesture recognition algorithms [9].

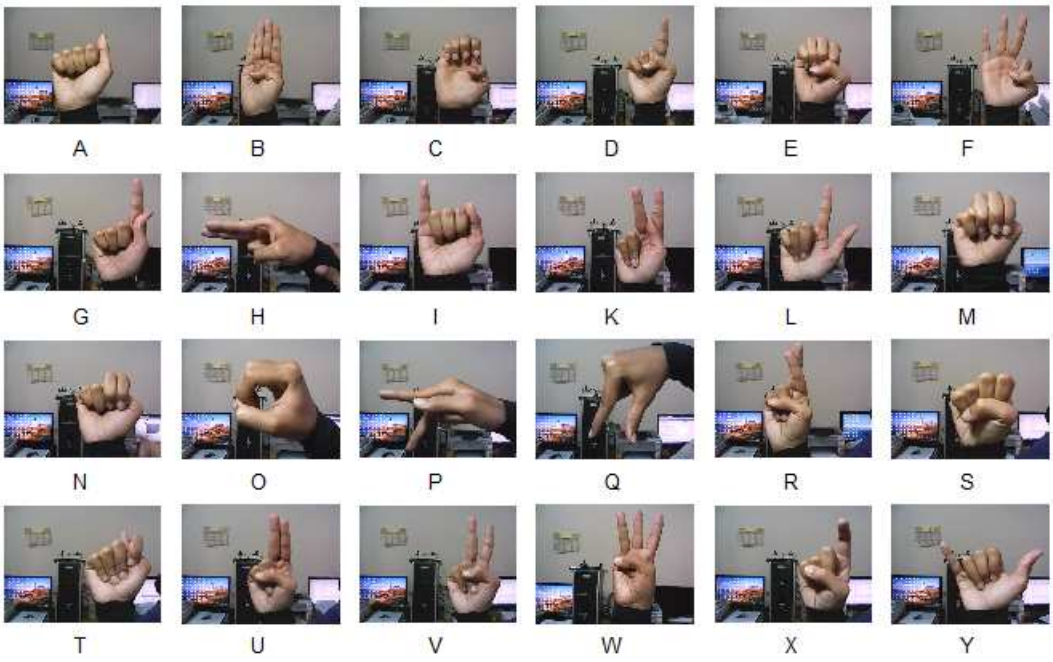


Figure 1.4: One static gesture image for each of 24 ASL hand alphabets with nonuniform background.

1.7 Organization of the Thesis

The thesis is organized as follows:

Chapter 1 presents introduction of the thesis work which includes review of the hand gestures and its applications, gesture recognition and associated techniques, motivation of work, literature survey, description of the hand gesture databases, and an outline of the work described in the thesis.

Chapter 2 describes the preprocessing steps of a gesture recognition system and proposes a technique to make the gesture image as rotation invariant. This chapter also proposes a combined feature set using LCS and block-based features for better representation of hand gesture images. Theoretical background of the homomorphic filtering, grayworld algorithm, Otsu segmentation, and Canny edge detection technique are also discussed in this chapter. This chapter also demonstrates the detailed methodology of the proposed technique. The experimental results are presented and discussed in detail.

Chapter 3 presents hand gesture recognition using DWT and F-ratio based fea-

ture set and RBF-NN classifier. Theoretical background of wavelet transform, F-ratio, Krawtchouk moments (KM), discrete cosine transform (DCT) and RBF neural network are also discussed here. In addition, the proposed framework of DWT and F-ratio based feature extraction technique which includes cropping and resizing of the image, applying DWT on resized and enhanced grayscale image and selection of DWT coefficient matrices using F-ratio, is also described in detail. The experimental results are shown and compared with earlier reported techniques.

Chapter 4 presents hand gesture recognition using modified RBF-NN classifiers, where the centers are automatically selected using k-mean algorithm and estimated weight matrix is updated using LMS algorithm to improve the recognition performance. Theoretical background of k-mean and LMS algorithms are also discussed here. In addition, the details of the two different proposed classifiers that are KLMS-RBFNN-GAF and k-mean and LMS-based RBF-NN classifier with composite sigmoidal activation function (represented as KLMS-RBFNN-SAF classifier), are described in this chapter. The experimental results are presented and compared with previously reported techniques.

Chapter 5 discusses feature vector optimization using genetic algorithm (GA) for hand gesture recognition. The details of GA is demonstrated here. The methodology for feature vector optimization is presented in detail. The performance of static hand gesture recognition using GA based optimized feature subset is also mentioned in this chapter. The experimental results are evaluated and compared with earlier reported techniques.

Chapter 6 makes the overall conclusions of the work and discusses the scope of future work in the same and further improvements.

CHAPTER 2

Preprocessing and Feature Extraction Techniques for Hand Gesture Recognition

2.1 Introduction

The human-computer interaction (HCI) or human alternative and augmentative communication (HAAC) has become an increasingly important part of our daily lives because of massive technological infusion in our lifestyle [13]. A suitable design of vision-based static hand gesture recognition technique can be used to develop an advanced hand gesture based interface system for successful HCI or HAAC applications [7, 8, 16]. Preprocessing is the first stage of a vision-based static hand gesture recognition system [12]. Preprocessing stage involves several sub-stages depending on the captured image condition and applied feature extraction techniques [7]. The preprocessing stage which is described in this chapter, consists of the following sub-stages: image enhancement which enhances the image by compensating illumination variation; segmentation, which segments hand region from its background image and transforms into binary silhouette; image rotation that makes segmented gesture as rotation invariant; filtering that effectively removes background noise and object noise from the binary image and provides a well defined segmented hand gesture. Feature extraction is an important stage for a hand gesture recognition system because it may not provide good recognition performance using the best classifier due to inefficient features. The objective of the feature extraction stage is to extract the most relevant features of the hand gesture image for recognition [7]. Features are selected based on either (1) best representation of a given class of signals, or (2) best distinction between classes [63]. The selection of good features can strongly affect the classification performance and reduce the computational time [7]. Various features have been reported in the literature [12,13,16,17,47–49,54,58] to represent static hand gesture image like statistical moments [13,47], block-based features [17], Haar-link feature [48], Fourier discriminator (FD) [53,54], lengths of vectors between the gesture's centroid to the useful portion of the gesture border [12], localized contour sequence (LCS) [16] and normalized silhouette distance signal features [49] etc. However, following benefits [16] of LCS features distinguish it from other features for hand gesture recognition: (i) The LCS is independent of shape complexity. (ii) The

LCS can also be used to efficiently characterize partial contours. (iii) The LCS representation is autonomous from derivative computations and it is effective with respect to contour noise. (iv) Increase window size enhances the magnitude of the LCS. The block based feature set [17] also have some important advantages: (i) This feature set incorporates the regional information of the hand gesture based on the partitions of the block and (ii) It is invariant in position and size of the gesture image. A large number of techniques [13, 17, 47, 53, 58] is reported by the researchers for recognition of static hand gestures. However, proper preprocessing and accurate feature representation of hand gesture images are still a challenge for static hand gesture recognition system in real-time application.

In the preprocessing stage, this work proposes an image rotation technique to make segmented gestures as rotation invariant by coinciding the first principal component of the segmented hand gestures with vertical axes. This work also proposes a combined feature set which couples LCS features with block-based features, to represent hand gesture image more accurately. Initially, two types of features are extracted from preprocessed hand gesture image: (i) LCS features (contour features) that efficiently represent and carry contour information of the object [64] and (ii) block-based features (regional features) that carry regional information of the object [17]. Then, the proposed combined feature set is obtained by appending the LCS features with the block-based features. The advantage of this combined feature set is, it contains the contour as well as regional information of the gesture and provides a better representation of hand gesture compared to each of the individual LCS or block-based feature set. The proposed combined feature set is applied to the input of the classifier to recognize the static hand gesture images. In this chapter, multilayer perceptron back propagation neural network (MLP-BP-NN) with single hidden layer is used as a classifier because MLP-BP-NN has following advantages: (i) it can be used to generate likelihood-like scores that are discriminative in the state level; (ii) it can be easily implemented in hardware platform for its simple structure; (iii) it has the ability to approximate functions and automatic similarity based generalization property; (iv) complex class distributed features

can be easily mapped by neural network [2, 65]. The performance of gesture recognition is tested on three hand gesture databases which include grayscale images with uniform background (Database I and Database II) and color images with nonuniform background (Database III). To investigate the performance of the proposed combined feature set and to compare it with LCS [16] and block-based [17] feature sets, experiments are separately conducted on three different static hand gesture databases using LCS, block-based and proposed combined feature sets using MLP-BP-NN as a classifier. Experimental results indicate that the proposed combined feature set provides better recognition performance compared to LCS [16] and block-based [17] feature sets individually. To investigate, the performance of the proposed image rotation technique, experiments are separately conducted on three different static hand gesture databases without image rotation and with image rotation techniques. The experimental results demonstrate that hand gesture recognition performance is significantly improved due to the proposed image rotation technique.

2.1.1 Organization of the Chapter

The rest of this chapter is organized as follows: Section 2.2 provides the basic concepts of homomorphic filtering, grayworld algorithm, Otsu segmentation, skin color segmentation and Canny edge detection. Section 2.3 describes the proposed methodology in details. Performance evaluation followed by a discussion are presented in the Section 2.4. Finally, the chapter concludes in Section 2.5.

2.2 Theoretical Background

2.2.1 Homomorphic Filtering

A homomorphic filtering technique is used to enhance an image by simultaneous gray-level range compression and contrast enhancement [66]. In illumination-reflectance

model [67], image pixel value $g(x, y)$ can be expressed as

$$g(x, y) \;=\; I(x, y)\, R(x, y) \tag{2.1}$$

where, $I(x, y)$ and $R(x, y)$ are illumination and reflection components respectively. The homomorphic filtering [66] technique is described as follows.

1. Take logarithm transform on input image to separate illumination and reflection components,

$$\begin{aligned} z(x, y) \;&=\; \ln g(x, y) \\ &=\; \ln I(x, y) + \ln\, R(x, y) \end{aligned} \tag{2.2}$$

2. Compute the frequency domain representation i.e. $Z(u, v)$ by taking Fourier transform on (2.2), therefore

$$\begin{aligned} \Im\{z(x, y)\} \;&=\; \Im\{\ln g(x, y)\} \\ &=\; \Im\{\ln\, I(x, y)\} + \Im\{\ln\, R(x, y)\} \end{aligned} \tag{2.3}$$

   or

$$Z(u, v) \;=\; F_I(u, v) + F_R(u, v) \tag{2.4}$$

   where, $Z(u, v)$, $F_I(u, v)$ and $F_R(u, v)$ are the Fourier transforms of $z(x, y)$, $ln\, I(x, y)$ and $ln\, R(x, y)$ respectively.

3. Obtain the filter output in frequency domain i.e. $S(u, v)$, by mean of filtering function $H_f(u, v)$

$$\begin{aligned} S(u, v) \;&=\; H_f(u, v) Z(u, v) \\ &=\; H_f(u, v) F_I(u, v) + H_f(u, v) F_R(u, v). \end{aligned} \tag{2.5}$$

4. Compute spatial domain representation, $s(x, y)$ by taking inverse Fourier trans-

form as presented in (2.6),

$$\begin{aligned} s(x,y) &= \Im^{-1}\{S(u,v)\} \\ &= \Im^{-1}\{H_f(u,v)F_I(u,v)\} \\ &\quad +\Im^{-1}\{H_f(u,v)F_R(u,v)\} \end{aligned} \tag{2.6}$$

where, $\Im^{-1}$ is the inverse Fourier transform operator. Lets assume

$$I^{'}(x,y) = \Im^{-1}\{H_f(u,v)F_I(u,v)\} \tag{2.7}$$

and

$$R^{'}(x,y) = \Im^{-1}\{H_f(u,v)F_R(u,v)\}. \tag{2.8}$$

Then, (2.6) can be expressed as

$$s(x,y) = I^{'}(x,y) + R^{'}(x,y). \tag{2.9}$$

5. Finally, the desired enhanced image $f(x,y)$ is obtained through exponential operation on (2.9),

$$\begin{aligned} f(x,y) &= e^{s(x,y)} \\ &= e^{I^{'}(x,y)}.e^{R^{'}(x,y)} \\ &= I_0(x,y)R_0(x,y) \end{aligned} \tag{2.10}$$

where, $I_0(x,y) = e^{I^{'}(x,y)}$ and $R_0(x,y) = e^{R^{'}(x,y)}$ are separately represent the illumination and reflectance components of the output image pixel.

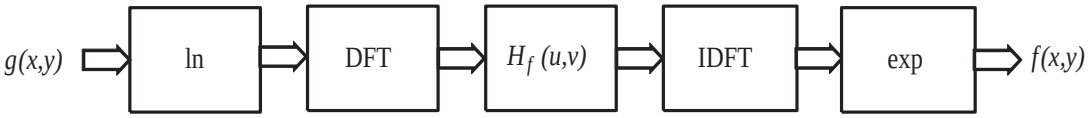


Figure 2.1: The operational block diagram of homomorphic filtering technique.

The block diagram of homomorphic filtering technique is shown in Figure 2.1 [67]. In this particular application, the key elements are the separation of illumination and

reflectance components from image pixel as indicated in (2.2) and operation of filtering function $H_f(u, v)$ on these components separately in frequency domain as indicated in (2.5). The filter function $H_f(u, v)$ [67] which is used here, defined as

$$H_f(u, v) = (a_H - a_L)[1 - e^{-c(D^2(u,v)/D_0^2)}] + a_L \tag{2.11}$$

where, $D(u, v)$ is the distance from point (u, v) to the origin of the frequency rectangle (frequency domain representation of image), D_0 is the cutoff distance measured from the origin and c is a constant, introduced to control the sharpness of the slope of the filter function shown in Figure 2.2 during its transition between a_L to a_H, where a_L and a_H represents the lowest and highest values of the filter function $H_f(u, v)$ respectively [66].

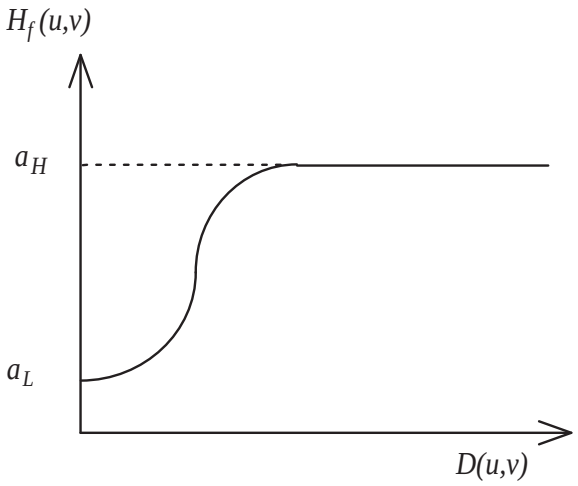


Figure 2.2: Radial cross section of a circularly symmetric homomorphic filter function.

In general, the illumination component of an image is described by slow spatial variations, while the reflectance component varies abruptly, particularly at the junction regions [66]. These characteristics lead to associate low and high frequencies of a logarithm image with illumination and reflectance components respectively [67]. These associations are utilized to normalize illumination variation of hand gesture images. If the

parameters a_L and a_H are chosen such that $a_L < 1$ and $a_H > 1$, the filter function shown in Figure 2.2 decreases the contribution made by the low frequencies (illumination) and increases the contribution made by high frequencies (reflectance) [66]. Therefore, the net result will contain the essential information of the image, simultaneously greatly reducing the influence of illumination variation.

2.2.2 Gray World Algorithm

The gray world algorithm [68] is a lighting compensation method that attempts to equalize the mean of the red (R), green (G), and blue (B) channels. This algorithm is developed on the basis of the gray world assumption, which indicates that the average intensity of the red, green, and blue channels should be equal for a typical scene. Let, an image $I(x, y)$ have spatial resolution $M \times N$ and let, $I_R(x, y)$, $I_G(x, y)$ and $I_B(x, y)$ denote the red, green, and blue channels of the image respectively, where x and y refer to the indices of the pixel position. First, the average of red channels (R_{AVG}), average of green channels (G_{AVG}), average of blue channels (B_{AVG}) and average of gray levels (μ_{AVG}) are calculated as follows:

$$\begin{aligned} R_{AVG} &= \tfrac{1}{MN} \sum_{x=1}^{M} \sum_{y=1}^{N} I_R(x, y) \\ G_{AVG} &= \tfrac{1}{MN} \sum_{x=1}^{M} \sum_{y=1}^{N} I_G(x, y) \\ B_{AVG} &= \tfrac{1}{MN} \sum_{x=1}^{M} \sum_{y=1}^{N} I_B(x, y) \\ \mu_{AVG} &= \tfrac{1}{3}(R_{AVG} + G_{AVG} + B_{AVG}) \end{aligned} \tag{2.12}$$

Then, the scale ratios of the original red, green and blue channels, A_R, A_G and A_B respectively are estimated by (2.13) [60].

$$\begin{aligned} A_R &= \mu_{AVG}/R_{AVG} \\ A_G &= \mu_{AVG}/G_{AVG} \\ A_B &= \mu_{AVG}/B_{AVG} \end{aligned} \tag{2.13}$$

Then, the modified red, green and blue channels of the image as represented $\widehat{I}_R$, $\widehat{I}_G$ and $\widehat{I}_B$ respectively are obtained by (2.14).

$$\begin{aligned} \widehat{I}_R(x,y) &= A_R.I_R(x,y) \\ \widehat{I}_G(x,y) &= A_G.I_G(x,y) \\ \widehat{I}_B(x,y) &= A_B.I_B(x,y) \end{aligned} \tag{2.14}$$

Now, the new estimated pixels of red, green and blue channels as represented by (2.14), satisfy the gray world assumption which compensates the lighting variation for a scene.

2.2.3 Otsu Segmentation

The Otsu segmentation [69] is a clustering based image thresholding technique. It is used to segment an image into two classes (foreground pixels and background pixels), or to transform an image into binary silhouette [2]. In Otsu segmentation algorithm, L gray level image is segmented into two classes $\Omega_0 = \{1, 2, \ldots, T\}$; and $\Omega_1 = \{T + 1, T + 2, \ldots, L\}$ using a threshold T. The optimum threshold T^* is determined as that value of T which maximizes the ratio between-class variance σ_B^2 to the total variance σ_T^2 [70]. If n_i is the number of pixels at i^{th} gray level and N is the total number of pixels, then for a given T, the σ_B^2 and the σ_T^2 are separately computed as in (2.15) and (2.16) respectively [70].

$$\sigma_B^2 = \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2 \tag{2.15}$$

$$\sigma_T^2 = \sum_{i=1}^{L} (i - \mu_T)^2 P_i \tag{2.16}$$

where

$$\omega_0 = \sum_{i=1}^{T} P_i, \quad \omega_1 = \sum_{i=T+1}^{L} P_i$$
$$\mu_0 = \sum_{i=1}^{T} (iP_i)/\omega_0, \quad \mu_1 = \sum_{i=T+1}^{L} (iP_i)/\omega_1$$
$$\mu_T = \sum_{i=1}^{L} (iP)$$
$$P_i = n_i/N, \left(P_i \geq 0 \text{ and } \sum_{\mathrm{i=1}}^{\mathrm{L}} P_i = 1 \right).$$

Here, P_i, μ_T, μ_0, μ_1, ω_0 and ω_1 represent the probability of occurrence in i^{th} gray level, total mean levels, class 0 mean levels, class 1 mean levels, probability of class 0 occurrence and probability of class 1 occurrence respectively. Typically, the entire histogram is scanned to find the optimum threshold T^*.

2.2.4 Canny Edge Detection Technique

The edge detection technique [71] extracts useful structural information from an image and significantly reduces the amount of data to be processed. The Canny edge detection is one of the most popular edge detection technique [71]. The general criteria for edge detection are as follow [72, 73]:

1. Detection: Edge should be detected with low error rate, which indicates that detection probability of real edge points should be maximized while detection probability of non-edge points as real-edge points should be minimized.

2. Localization: The detected edges should be as close as possible to the real edges.

3. Number of responses: One real edge should be marked only once and should not be created false edges due to image noise.

The Canny edge detection technique consists of the following five steps [73]: (a) Smoothing: smoothing the image using Gaussian filter to remove noise; (b) Finding gradients: find the intensity gradients of the image for marking the edges; (c) Non-maximum suppression: deleting all the gradient values in the gradient image except the local maximum which should be marked as edges; (d) Double thresholding: potential

edges are determined using double thresholding and (e) Edge tracking by hysteresis: final edges are determined by suppressing all other edges that are weak and not connected to strong edges. The details of each step [72] are given as follows.

(a) Smoothing: In general, all images contain some amount of noise during acquisition from camera. To prevent that noise is mistaken for edges, noise must be reduced. Since the Gaussian filter can be computed using a simple mask, it is used in the Canny algorithm to smooth the image [73]. The smoothing can be performed using standard convolution of the actual image with Gaussian mask. The kernel of a Gaussian filter with a standard deviation $\sigma = 1.4$ for 5×5 Gaussian mask, K_B [72] is shown in (2.17).

$$K_B = \frac{1}{159}. \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} \tag{2.17}$$

(b) Finding gradients: The Canny algorithm basically finds edges where the grayscale intensity of the image changes the most. These areas are found by determining gradients of the image. Gradients at each pixel in the smoothed image are determined by applying the Sobel-operator [67]. First step is to approximate the gradient of smoothed image in the x-direction (K_{G_x}) and y-direction (K_{G_y}) respectively by applying the kernels [67] as presented in (2.18).

$$K_{G_x} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, K_{G_y} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{2.18}$$

Then, the magnitude and direction of gradient (i.e. edge strength and direction of the edges) can be determined using (2.19) and (2.20) respectively.

$$|G| = \sqrt{G_x^2 + G_y^2} \tag{2.19}$$

$$\theta = \arctan\left(\frac{|G_y|}{|G_x|}\right) \tag{2.20}$$

where G_x, G_y, $|G|$ and θ represent the gradient in the x- direction, the gradient in y-direction, the magnitude of gradient and the direction of gradient, respectively.

(c) Non-maximum suppression: The edge extracted from the gradient value is still quite blurred. Therefore, non-maximum suppression [74] is applied to convert the "blurred" edges in the image to "sharp" edges. This is done by deleting everything except the local maximum which should be preserved as edges in the gradient image. The algorithm for each pixel in the gradient image is as follows.

1. The gradient direction θ is rounded to nearest 45°, corresponding to the use of an 8-connected neighborhood.

2. The edge strength of the current pixel is compared with the edge strength of the pixel in the positive and negative gradient direction. For example, if the gradient direction is north (i.e $\theta = 90°$), edge strength of the current pixel is compared with the pixels to the north and south.

3. If the edge strength of the current pixel is largest compared to the pixel in the positive and negative gradient direction, then preserve the value of the edge strength. If not, suppress the value.

(d) Double thresholding: The edge pixels are quite accurate to present the real edges after the non-maximum suppression. However, there are still some edge pixels caused by noise or color variations. The Canny edge detection technique uses double thresholding [74] to clarify the different types of edge pixels. If the gradient values of edge pixels are higher than the high threshold value, they are marked as strong edges. If the gradient values of edge pixels are lower than the low threshold value, they will be suppressed. If the gradient values of edge pixels are between the low and high threshold values, they are marked as weak edges.

(e) Edge tracking by hysteresis: The strong edges can immediately be included in the final edge image, as they are extracted from the true edges in the image. Since

the weak edges can either be due to true edges or noise/color variations, the weak edges are included in the final edge image if and only if they are connected to strong edges. To track the edge connection at a weak edge pixel and its 8-connected neighborhood pixels, Binary Large Object-analysis (BLOB) [72] is applied after double thresholding. For a weak edge, if at least one strong edge pixel is involved in the BLOB, then the weak edge point can be identified as true edge and included in the final edge image.

2.3 Methodology

This chapter clearly describes the methodology followed for preprocessing and feature extraction techniques. In the preprocessing stage, an image rotation technique is proposed to make the segmented image as rotation invariant using the direction of the first principal component. A combined feature set is also proposed here for better representation of hand gesture image. The different stages of a static hand gesture recognition method are given as follows.

2.3.1 Preprocessing

The preprocessing stage for a hand gesture recognition method involves the following steps: image enhancement, segmentation, rotation and morphological filtering. Figure 2.3 and Figure 2.4 show the results of different preprocessing steps for a grayscale and color hand gesture images respectively.

2.3.1.1 Enhancement

The aim of image enhancement is to process an image so that the illumination variation within the image is normalized and the resultant image is more suitable for segmentation. For grayscale images of Databases I and II, this work uses homomorphic filtering [67] technique to enhance hand gesture images by normalizing illumination variation within it. For color images of Database III, this work utilizes gray world algorithm [68] to enhance gesture images by compensating the variation of light. Figure 2.3

(b) and Figure 2.4 (b) represent the results of image enhancement done by homomorphic filtering and gray world technique respectively.

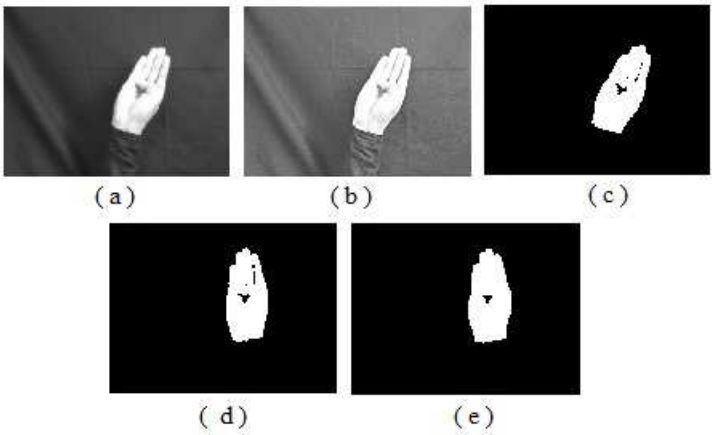


Figure 2.3: Results of preprocessing stage for a grayscale hand gesture image of Database II. (a) Original gesture image. (b) Enhanced image. (c) Otsu segmented image. (d) Rotated image. (e) Morphological filtered image.

2.3.1.2 Segmentation

The objective of the segmentation technique is to extract the hand region from the background of the gesture image. For grayscale images of Database I and II, the hand region of the enhanced gesture image is segmented using Otsu segmentation method [70]. The result of Otsu segmentation method is shown in Figure 2.3 (c). For color images of Database III, the hand is segmented from the background of the enhanced gesture image using skin color detection method in YCbCr color space [75]. In contrast to RGB, the YCbCr color space gives better performance under varying lighting condition because of the luma-independent characteristic of Cb and Cr. In this work, a pixel is considered as skin pixel if $Th < Y < 255$, $85 < Cb < 128$ and $129 < Cr < 185$, where Th is (1/3) of average Y value of all pixels. The ranges of Cb and Cr are selected using the histogram of Cb and Cr for manually cropped hand gesture regions [76]. The skin color detection result for hand gesture image is shown in Figure 2.4 (c). From the detection result, it is observed that the skin color detection image also contains other skin color objects not

belonging to hand region. Therefore, the hand region is assumed to be largest connected skin color object, and other detected skin color objects are filtered out by comparing their area. The final segmented output is shown in Figure 2.4 (d).

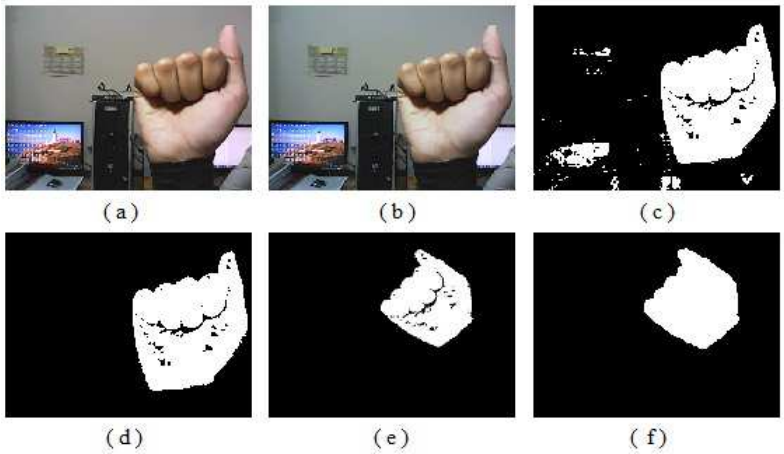


Figure 2.4: Results of preprocessing stage for a color hand gesture image of Database III. (a) Original gesture image. (b) Enhanced image. (c) Detected skin color region. (d) Final segmented image. (e) Rotated image. (f) Morphological filtered image.

2.3.1.3 Rotation

In this work, principal component analysis (PCA) approach is utilized to rotate segmented gesture image. Mathematically PCA is an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate (known as first principal component), the second greatest variance on the second coordinate, and so on [77]. In this study, an image rotation technique is proposed to make the segmented image as rotation invariant using the direction of the first principal component. A stepwise description of the proposed technique is as follows.

1. First, represent all nonzero pixel position of segmented gesture as

$$H = [X_1, X_2, X_3, \cdots, X_M] \tag{2.21}$$

where, H is the position vector, $X_i = [x_{i1},\, y_{i1}]^T$ is the position of i^{th} nonzero pixel with respect to the input coordinate system and M is the total number of nonzero pixel in segmented gesture.

2. Compute covariance matrix (C) using (2.22)

$$\mathrm{C} = \left(H - \bar{X}\right)\left(H - \bar{X}\right)^T \tag{2.22}$$

where $\bar{X} = [\bar{x}_1,\, \bar{y}_1]^T$ is the centroid of the segmented gesture with respect to the input coordinate system.

3. Compute eigenvalues and eigenvectors of C and find the first principal component. Let, the eigenvalues of C be λ_1 and λ_2 and corresponding eigenvectors be $e_1 = [e_{11},\, e_{21}]^T$ and $e_2 = [e_{21},\, e_{22}]^T$. Since $\lambda_1 > \lambda_2$, $e_1 = [e_{11},\, e_{21}]^T$ is the first principal component of segmented gesture.

4. Compute the direction of the first principal component by (2.23)

$$direction = \tan^{-1}\left(\frac{e_{21}}{e_{11}}\right) \tag{2.23}$$

5. Find out the rotation angle between the first principal axes of the segmented hand gesture and vertical axes.

6. Finally, rotate the segmented hand gesture, so that the first principal axes of the segmented hand gesture coincide with vertical axes.

The results of proposed image rotation technique for a hand gesture image of Databases II and III are shown in Figs. 2.3 (d) and 2.4 (e) respectively.

2.3.1.4 Filtering

A morphological filtering [78] technique is applied here to reduce the object noise and to obtain a well defined smooth, closed, and complete segmented hand gesture and this is done using a sequence of dilation and erosion operations over the rotation invariant

segmented hand gesture image. In general, the dilation and erosion operations on a binary image P_1, with a structuring element Q_1 are defined [67] as follows.

Dilation: If P_1 and Q_1 are sets in the 2-D integer space Z^2; $x = (x_1, x_2)$ and ϕ is the empty set, then, the dilation of P_1 by Q_1 is defined as

$$P_1 \oplus Q_1 = \{x \mid (\hat{Q_1})x \cap P_1 \neq \phi\}$$

where, $\hat{Q_1}$ is the reflection of Q_1. Dilation consists of two steps i.e. obtaining the reflection of Q_1 about its origin and then shifting this reflection by x. The dilation of P_1 by Q_1 is the set of all x displacements such that $\hat{Q_1}$ and P_1 overlap by at least one nonzero element. Set Q_1 is commonly referred as the structuring element.

Erosion: The erosion of P_1 by Q_1 is defined as

$$P_1 \ominus Q_1 = \{x \mid (Q_1)x \subseteq P_1\}$$

it indicates that the erosion of P_1 by Q_1 is the set of all points x such that Q_1, translated by x, is contained in P_1. Note that dilation expands an image and erosion shrinks it.

Opening: The opening of P_1 by Q_1 is defined as

$$P_1 \circ Q_1 = (P_1 \ominus Q_1) \oplus Q_1$$

Thus, the opening of P_1 by Q_1 is simply the erosion of P_1 by Q_1 followed by a dilation of the result by Q_1. Opening generally smooths the contour of an image, breaks narrow isthmuses, and eliminates thin protrusions.

Closing: The closing of P_1 by Q_1 is is defined as

$$P_1 \bullet Q_1 = (P_1 \oplus Q_1) \ominus Q_1$$

This says that the closing of P_1 by Q_1 is simply the dilation of P_1 by Q_1, followed by the erosion of the result by Q_1. Closing also tends to smooth sections of contour but, as opposed to opening, it generally fuses narrow breaks and long thin gulfs, eliminates

small holes, and fills gaps in the contour. The outputs of morphological filtering for a hand gesture image of Databases II and III are shown in Figs. 2.3 (e) and 2.4 (f), respectively.

2.3.2 Feature Extraction

Feature is a distinctive or characteristic measurement, transform, structural component extracted from a segment of a pattern. Features are used to represent patterns with the goal of minimizing the loss of important information [63]. The aim of the feature extraction stage is to capture and discriminate the most relevant characteristics of the hand gesture image for recognition [7]. The selection of good features can strongly affect the classification performance and reduce the computational time. The features used must be suitable for the application and the applied classifier [7]. In this chapter, localized contour sequence (LCS) features [16], block based features [17] and a proposed combined features are extracted to represent the hand gesture. The details of above feature extraction techniques are given as follows.

2.3.2.1 Localized Contour Sequence (LCS) Features

The shape of the contour can be used to identify hand gestures [79]. Therefore, the LCS which efficiently represents the object's contour [64] is selected as a feature set for hand gesture. A well established canny edge detection algorithm [71] is used to detect the edge of the preprocessed hand gesture. In this work, a contour tracking algorithm is used to track the contour of the edge detected gesture in the clockwise direction starting from the topmost left contour pixel. The contour tracking algorithm is given in Algorithm 2.1 [2]. By computing the contour tracking algorithm, the positions of all contour pixel are stored sequentially into x and y arrays. If $h_i = (x_i, y_i)$, $i = 1, 2 \ldots, N$ is the i^{th} contour pixel in the sequence of N ordered contour pixels of a gesture, the i^{th} sample $h(i)$ of the LCS [64] of the gesture is obtained by computing the perpendicular Euclidean distance between h_i and the chord connecting the end-points $h_{[i-(w-1)/2]}$ and $h_{[i+(w-1)/2]}$ of a window of size w boundary pixels centered on h_i. An odd integer is

Algorithm 2.1: : The contour tracking algorithm.

1: The image is scanned from top to bottom and left to right to find first contour pixel being marked as $P_1(i, j)$.
2: The position of P_1 is stored into x and y arrays respectively (i.e., $x[1] = i$ and $y[1] = j$). Initially, set $i_1 = 0, i_2 = 0, j_1 = 0, j_2 = 0$ and start search for next contour pixel.
3: P_J is searched by scanning clockwise direction in the sequence of $P_J(i, j-1), P_J(i-1, j-1), P_J(i-1, j), P_J(i-1, j+1), P_J(i, j+1), P_J(i+1, j+1), P_J(i+1, j)$ and $P_J(i+1, j-1)$ until $P_J = 1$. Where $J = 2, 3, 4, \ldots, N$ and N is total number of detected contour pixel.
4: If $P_J = 1$, position of $P_J \neq (i_1, j_1)$ and position of $P_J \neq (i_2, j_2)$, position of P_J is stored separately into x and y arrays.
5: Set (i, j) = position of P_J and when $J > 3$, set (i_1, j_1) = position of P_{J-1} and (i_2, j_2) = position of P_{J-2}.
6: If step (4) becomes false then set $x[J] = x[J-1], y[J] = y[J-1], (i, j)$ = position of $P_{J-1}, (i_1, j_1)$ = position of P_J and (i_2, j_2) = position of P_{J-2}.
7: Repeat steps (2)-(5) until position of P_J = position of P_1.

chosen as the window size w. The LCS is computed as in (2.24)

$$h(i) = |u_i / v_i|, \tag{2.24}$$

where

$$\begin{aligned} u_i = x_i[y_{i-(w-1)/2} - y_{i+(w-1)/2}] \\ + y_i[x_{i+(w-1)/2} - x_{i-(w-1)/2}] \\ + [y_{i+(w-1)/2}][x_{i-(w-1)/2}] \\ - [y_{i-(w-1)/2}][x_{i+(w-1)/2}], \end{aligned}$$

and

$$\begin{aligned} v_i = [(y_{i-(w-1)/2} - y_{i+(w-1)/2})^2 \\ + (x_{i-(w-1)/2} - x_{i+(w-1)/2})^2]^{.1/2} \end{aligned}$$

The LCS is used to represent hand gesture contours because it has the following important properties [16]: (i) The LCS is not limited by shape complexity and it is, therefore, appropriate for gestures which typically have convex and concave contours. (ii) The LCS has been used to represent partial contours robustly. Therefore, the gesture representation will not be affected if a part of the gesture is obscured. (iii) No derivative

operation is involved during computations, therefore, the representation is quite robust with respect to contour noise (random variations in the contour). (iv) The amplitudes of the samples of the localized contour sequence proportionally increase with w. An increase in the amplitudes enhances the signal-to-noise ratio for a fixed contour noise level. In this work, 45 is empirically chosen as the size of w [2].

The duration and amplitude of the LCS vary from gesture to gesture. Therefore, these are normalized by setting its duration as the average LCS duration of training dataset and standard deviation as unity. In this work, the LCS durations are normalized as 150, 300 and 400 for Databases I, II and III respectively. Note that the normalized LCS feature set is position and size invariant of gesture images. Figure 2.5 shows the detected edge and corresponding normalized LCS features of a preprocessed static hand gesture image.

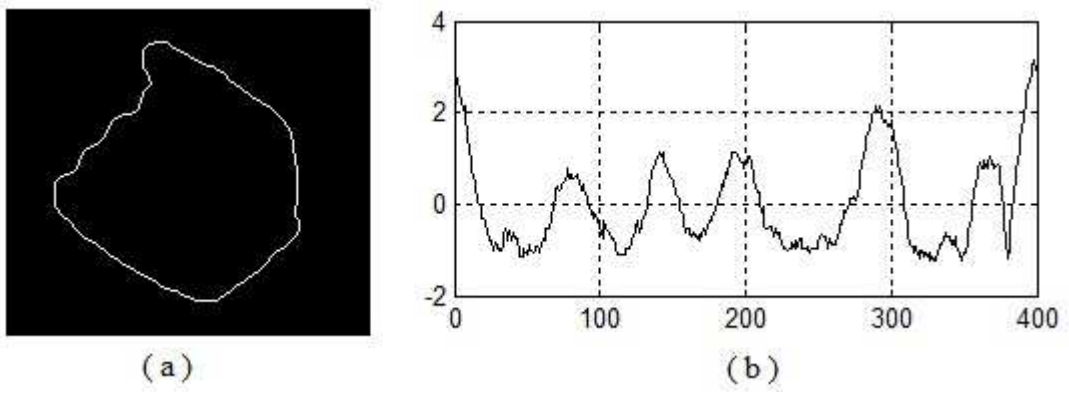


Figure 2.5: Detected edge and the normalized LCS features of a preprocessed static hand gesture image. (a) Detected edge of the preprocessed gesture image (b) Normalized LCS features for corresponding image.

2.3.2.2 Block-based Features

A bounding box is constructed around the hand region of the preprocessed gesture image [17]. The box is then cropped and partitioned into blocks. A block-based feature vector of the image incorporates the aspect ratio of the cropped bounding box, and the average intensity of each block. The block-based feature vector of length $V = 1+B_r \times B_c$,

denoted as $F_b = [f_1, \ldots, f_i, \ldots, f_V]$, where B_r and B_c are the number of rows and columns of the block partition. The first feature of F_b represents the aspect ratio of the bounding box and the remaining features represent block averages indexed row-wise from left to right. The bounding box and a restriction on the height of the static gesture make the block-based feature vector position and size invariant. The parameters, B_r and B_c are chosen through heuristic approach [17]. In this work, the static hand gestures are represented by feature vectors of length 13 i.e.$(1 + 4 \times 3)$ for Databases I and II and 21 i.e.$(1 + 5 \times 4)$ for Database III. Figure 2.6 shows the artwork of block-based feature extraction technique for a gesture image.

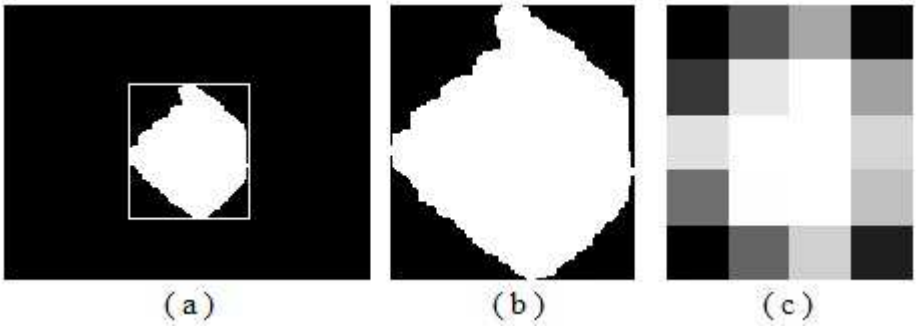


Figure 2.6: Artwork of block-based feature extraction technique: (a) hand gesture within bounding box, (b) cropped bounding box region and (c) 5×4 block partition.

2.3.2.3 Combined Features

This work proposes a combined features concept to represent hand gesture. Combined features are obtained by appending the LCS features with the block-based features. The block diagram of combined features extraction technique is shown in Figure 2.7. Assume, the LCS feature vector L_{lcs} is represented as $L_{lcs} = [L_1, L_2, L_3, \ldots, L_M]$ and the block-based feature vector F_b is represented as $F_b = [f_1, f_2, f_3, \ldots, f_V]$, where M and V represent the dimensions of LCS feature vector and block-based feature vector respectively. Therefore, the proposed combined feature vector $F_{combined}$ can be represented

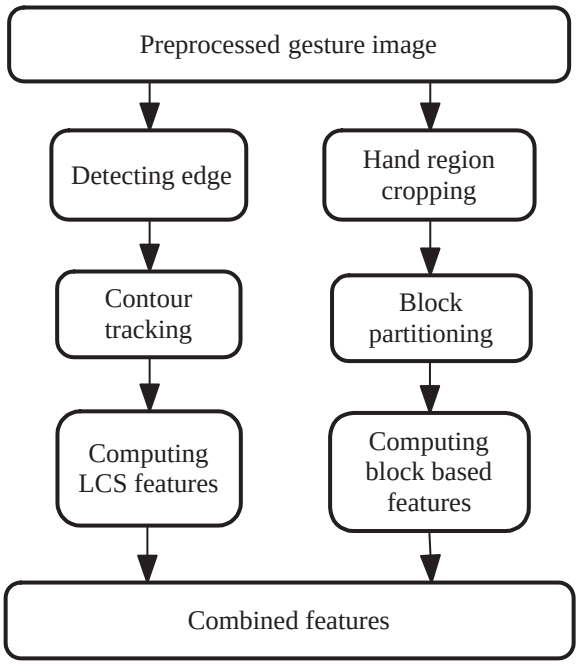


Figure 2.7: Block diagram of proposed combined features extraction technique.

as

$$F_{combined} = [L_1, L_2, L_3, \ldots, L_M, f_1, f_2, f_3, \ldots, f_V].$$

The combined feature set incorporates all the advantages of both LCS and block-based feature sets. It carries contour as well as regional information of the gesture image. Therefore, it provides a better representation of the static hand gesture image compared to each of the individual LCS or block-based feature set.

2.3.3 Classification: Artificial Neural Network

Classification is the final stage of a static hand gesture recognition system. Artificial neural networks (ANNs) are widely recognized as powerful classification methods that have been used extensively to solve many problems in different domains [7, 80]. In this study, multilayer perceptron back propagation neural network (MLP-BP-NN) with single hidden layer, which is one of the most popular supervised neural network models, is used to classify hand gestures. The basic structure of a MLP-BP-NN classifier with single

hidden layer is shown in Figure 2.8, where, m, h and n represent the number of input, hidden and output nodes of the network respectively. In MLP-BP-NN, the number of input nodes is equal to the number of input features. In output layer, the number

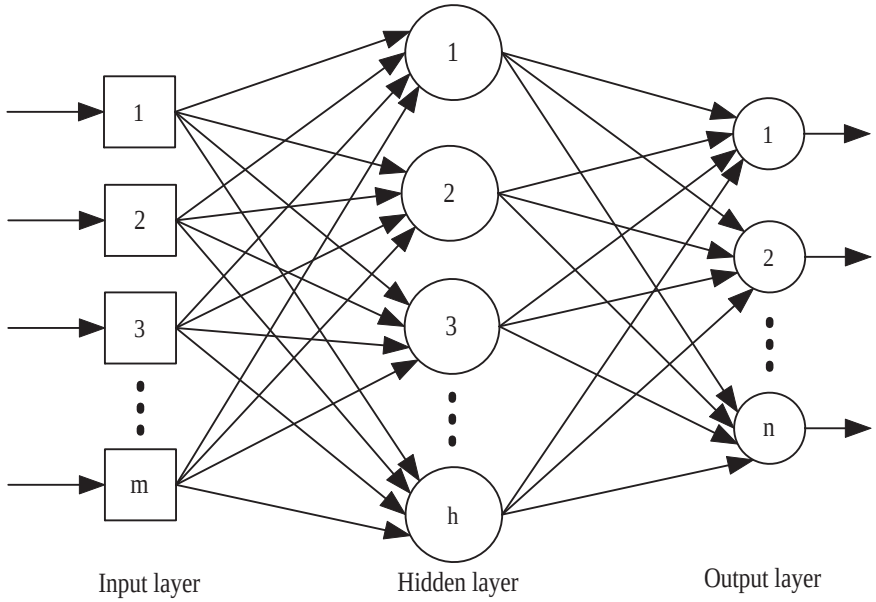


Figure 2.8: The basic structure of a MLP-BP-NN classifier with single hidden layer.

of output nodes (or neurons) represents the number of hand gesture types that to be classified. The number of hidden nodes is empirically chosen based on the classification performance. Each layer is fully connected with previous layer, and has no other extra connection. The MLP-BP-NN is trained using back propagation learning algorithm [80]. The back propagation learning algorithm works on the principle of generalized delta learning rule with an iterative gradient algorithm designed to minimize the root mean square (RMS) error between the actual and desired outputs of a MLP-BP-NN. There are several activation functions such as logistic function, identity function and hyperbolic tangent function that can be used in MLP-BP-NN classifier. The hyperbolic tangent function is chosen empirically in this application. The randomly selected initial weight and bias values of the MLP-BP-NN are updated with empirically selected 0.5 learning rate [2].

2.4 Performance Evaluation

In this chapter, the performance of gesture recognition is tested on three static hand gesture databases which include grayscale gesture images with uniform background (Databases I and II) and color gesture images with nonuniform background (Database III). Databases I consists of 1000 uniform background grayscale images of 25 static gesture with spatial resolution 256×248. Database II consists of 2400 uniform background grayscale images of ASL hand alphabet for 24 static gesture with spatial resolution 320×240. Each ASL hand alphabet contains 100 images; 10 samples are performed by 10 volunteers against an uniform black background based on the different angle, position and distance from the camera under different lighting conditions. Database III contains 2400 nonuniform background color images of ASL hand alphabet for 24 static gesture with spatial resolution 320×240. Each ASL hand alphabet comprises of 100 images; 10 samples are performed by 10 volunteers against a nonuniform background with variations of angle, position and size changes in different lighting conditions [81].

The gesture recognition performance is evaluated on the basis of four statistical indices [82] including classification accuracy (Acc), sensitivity (Sen), positive predictivity (Ppr) and specificity (Spe). Recognition accuracy (Acc) measures the overall system performance over all classes of gestures. It is the ratio of correctly classified gestures to the total number of gesture classified.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \tag{2.25}$$

where TP, TN, FP and FN are known as true positives, true negatives, false positives and false negatives respectively.

Sensitivity (Sen) or true positive rate (Tpr) is the ratio of correctly classified gesture among all gestures.

$$Sen = \frac{TP}{TP + FN} \times 100 \tag{2.26}$$

Positive predictivity (Ppr) is the ratio of the number of correctly detected gestures

TP, to the total number of gestures detected by the analyzer and is given as

$$Ppr = \frac{TP}{TP + FP} \times 100 \tag{2.27}$$

Specificity (Spe) is the ratio of the number of correctly rejected non-gestures TN, to the total number of non-gestures and is given as

$$Spe = \frac{TN}{TN + FP} \times 100 \tag{2.28}$$

In the present chapter, the static hand gesture recognition performances are evaluated based on two techniques (recognition without image rotation and recognition with image rotation) using LCS, block-based and proposed combined feature sets. The dimension of used LCS, block-based and proposed combined feature sets are 150, 13 and 163 respectively for Database I, 300, 13 and 313 respectively for Database II and 400, 21 and 421 respectively for Database III. For both techniques, a single hidden layer MLP-BP-NN classifier is used to classify the hand gesture images based on their feature sets.

2.4.1 Hand Gesture Recognition Without Image Rotation

To investigate hand gesture recognition performance without image rotation technique, each database is partitioned into training and testing sets. Randomly selected 50% image of all gestures are used for training and the remaining 50% image are utilized for testing. The experiments are separately conducted on three distinct databases utilizing LCS [16], block-based [17] and proposed combined feature sets with MLP-BP-NN classifier to evaluate the recognition performance for each feature set. The comparative performances of gesture recognition without image rotation technique using LCS [16], block-based [17] and proposed combined feature sets are tabulated in Table 2.1. The comparative results indicate that the average accuracy, sensitivity, positive predictivity and specificity of hand gesture recognition without image rotation technique using proposed combined feature set are 99.94%, 99.20%, 99.24% and 99.97% respectively for Database I, 98.82%, 85.83%, 86.15% and 99.38% respectively for Database II and

99.74%, 96.92%, 96.99% and 99.87% respectively for Database III. On the other hand, the average accuracy, sensitivity, positive predictivity and specificity of hand gesture recognition without image rotation technique using LCS features are 99.90%, 98.80%, 98.89% and 99.95% respectively for Database I, 98.76%, 85.17%, 85.48% and 99.36% respectively for Database II and 99.67%, 96.08%, 96.14% and 99.83% respectively for Database III, and using block-based features are 99.78%, 97.20%, 97.25% and 99.88% respectively for Database I, 97.55%, 70.58%, 70.37% and 98.72% respectively for Database II and 99.31%, 91.67%, 91.68% and 99.64% respectively for Database III. In this exper-

Table 2.1: Comparative performances of hand gesture recognition without image rotation technique using LCS, block-based and proposed combined feature sets for three distinct databases.

Database	Feature set	Acc (%)	Sen (%)	Ppr (%)	Spe (%)
Database I	LCS [16]	99.90	98.80	98.89	99.95
	Block-based [17]	99.78	97.20	97.25	99.88
	Proposed combined	**99.94**	**99.20**	**99.24**	**99.97**
Database II	LCS [16]	98.76	85.17	85.48	99.36
	Block-based [17]	97.55	70.58	70.37	98.72
	Proposed combined	**98.82**	**85.83**	**86.15**	**99.38**
Database III	LCS [16]	99.67	96.08	96.14	99.83
	Block-based [17]	99.31	91.67	91.68	99.64
	Proposed combined	**99.74**	**96.92**	**96.99**	**99.87**

iment, a single hidden layer MLP-BP-NN is used as a classifier to recognize the hand gesture images based on their feature sets. A neural network with a small number of neurons may not be sufficiently powerful to model a complex function [83]. On the other hand, a neural network with too many neurons may lead to overfitting of the training sets and lose its ability to generalize which is the primary desired characteristic of a neural network [83]. The recognition performance of the classifier is dependent on the number of hidden nodes in hidden layer [2]. However, there is no technique available for the exact selection of hidden nodes in a MLP-BP-NN [65]. In this work, the numbers of hidden nodes of the MLP-BP-NN are empirically selected based on best recognition performance. As an example, for Database II, the variation of gesture recognition per-

formance with respect to the number of hidden nodes using proposed combined feature set is shown in Figure 2.9. It is observed from Figure 2.9 that the best recognition performance is achieved when the number of hidden nodes in single hidden layer MLP-BP-NN classifier is 200. The number of input and output nodes of the MLP-BP-NN are selected based on the length of applied feature set and number of hand gesture classes respectively.

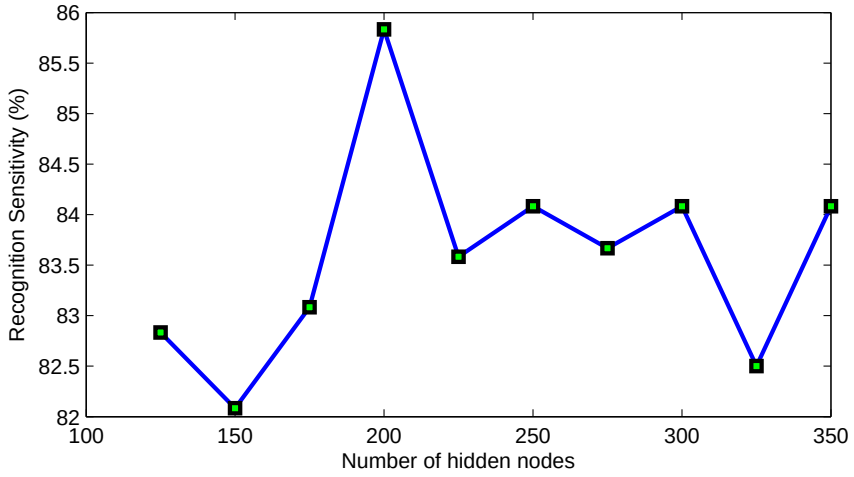


Figure 2.9: The performances of gesture recognition using proposed combined feature set for Database II when different number of hidden nodes are used in a single hidden layer MLP-BP-NN.

The performance of hand gesture recognition without image rotation technique using LCS, block-based and proposed combined features are further analyzed in terms of receiver operating characteristic (ROC) graph [84]. The relationship between sensitivity and specificity are described by the receiver operating characteristic (ROC) graph which alleviates improved analysis in terms of the recognition performance of a recognition technique. On an ROC graph, false positive rate (FPR) (or 1-specificity) and true positive rate (TPR) (or sensitivity) are separately plotted on the X and Y axis. The TPR and the FPR are separately defined by (2.26) and (2.29) respectively.

$$Fpr = \frac{FP}{TN + FP} \times 100 \tag{2.29}$$

For accurate recognition, TPR=1 and FPR=0 corresponds to the upper left corner of the ROC graph. Therefore, in ROC graph, the combination of TPR-FPR is considered better when it is located nearer to the upper left corner. Figure 2.10 shows ROC graphs of hand gesture recognition without image rotation technique using block-based, LCS and proposed combined features for three district databases. It is observed that recognition without image rotation technique using proposed combined feature set provides a higher TPR but lower FPR compared to LCS and block-based feature sets, individually for all three databases. Therefore, from Table 2.1 and Figure 2.10, it is observed that

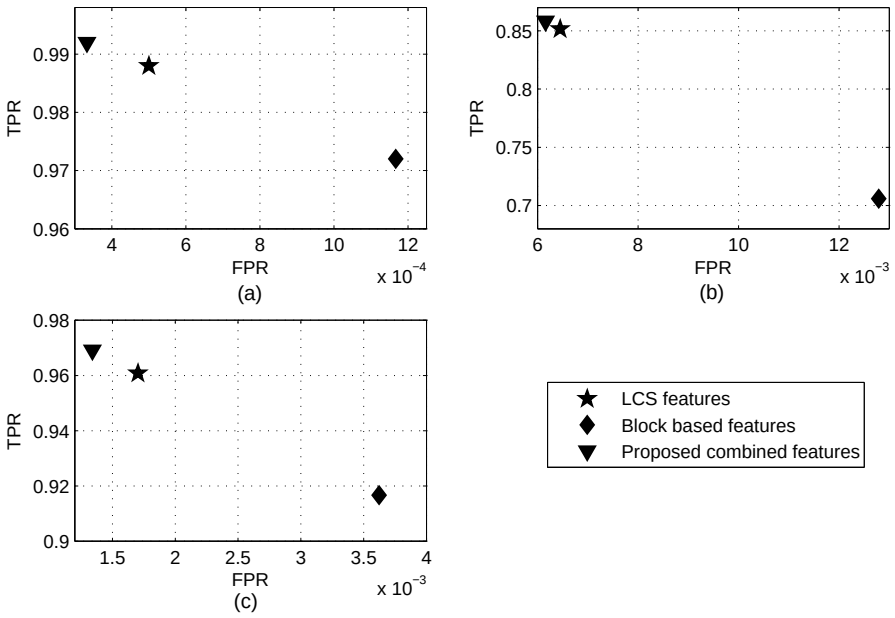


Figure 2.10: ROC graphs of hand gesture recognition without image rotation technique using LCS, block-based and proposed combined features for (a) Database I, (b) Database II and (c) Database III.

proposed combined feature set offers better recognition performance compared to LCS and block-based features, individually for all three databases.

2.4.2 Hand Gesture Recognition With Image Rotation

To investigate hand gesture recognition performance with image rotation technique, experiments are separately conducted on three distinct databases utilizing LCS [16], block-based [17] and proposed combined feature sets with MLP-BP-NN classifier. The comparative results are tabulated in Table 2.2. The comparative results show that the average accuracy, sensitivity, positive predictivity and specificity of hand gesture recognition with image rotation technique using LCS features are 99.94%, 99.20%, 99.26% and 99.97% respectively for Database I, 99.26%, 91.08%, 91.25% and 99.61% respectively for Database II and 99.75%, 97.00%, 97.07% and 99.87% respectively for Database III, and using block-based features are 99.89%, 98.60%, 98.70% and 99.94% respectively for Database I, 98.35%, 80.17%, 80.16% and 99.14% respectively for Database II and 99.40%, 92.83%, 92.85% and 99.69% respectively for Database III. On the other hand, the average accuracy, sensitivity, positive predictivity and specificity of hand gesture recognition with image rotation technique using proposed combined feature set are 99.95%, 99.40%, 99.43% and 99.98% respectively for Database I, 99.39%, 92.67%, 92.75% and 99.68% respectively for Database II and 99.79%, 97.50%, 97.55% and 99.89% respectively for Database III. In this study, the numbers of hidden nodes of the MLP-BP-NN

Table 2.2: Comparative performances of hand gesture recognition with image rotation technique using LCS, block-based and proposed combined feature sets for three distinct databases

Database	Feature set	Acc (%)	Sen (%)	Ppr (%)	Spe (%)
Database I	LCS [16]	99.94	99.20	99.26	99.97
	Block-based [17]	99.89	98.60	98.70	99.94
	Proposed combined	**99.95**	**99.40**	**99.43**	**99.98**
Database II	LCS [16]	99.26	91.08	91.25	99.61
	Block-based [17]	98.35	80.17	80.16	99.14
	Proposed combined	**99.39**	**92.67**	**92.75**	**99.68**
Data base III	LCS [16]	99.75	97.00	97.07	99.87
	Block-based [17]	99.40	92.83	92.85	99.69
	Proposed combined	**99.79**	**97.50**	**97.55**	**99.89**

are also empirically selected based on best recognition performance. As an example, the variation of recognition performance with respect to number of hidden nodes using proposed combined feature set for Database II is shown in Figure 2.11. It is seen from Figure 2.11 that the best recognition performance is achieved when number of hidden nodes in single hidden layer MLP-BP-NN classifier is 220.

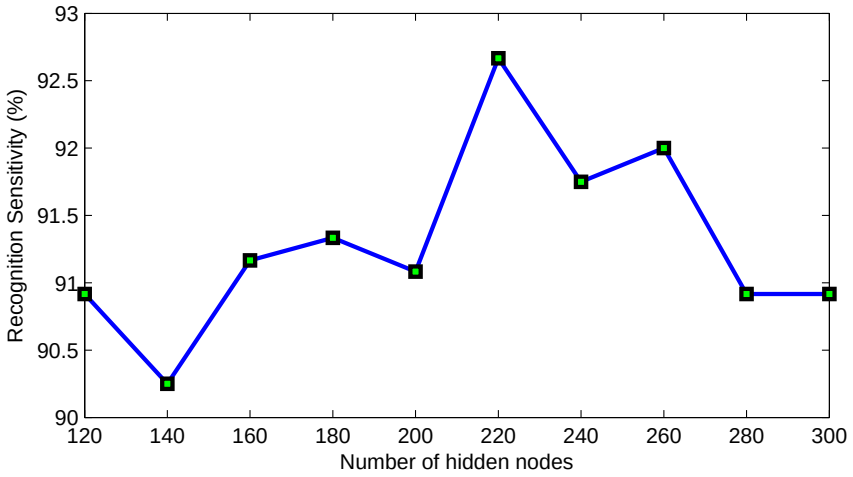


Figure 2.11: The performances of gesture recognition using proposed combined feature set for Database II with the different number of hidden nodes in a single hidden layer MLP-BP-NN.

The hand gesture recognition performances with image rotation technique using LCS, block-based and proposed combined features are also analyzed using ROC graph. The ROC graphs of hand gesture recognition with image rotation technique using block-based, LCS and proposed combined features for three distinct databases are shown in Figure 2.12. Table 2.2 and Figure 2.12 indicate that the recognition performance with image rotation technique using the proposed combined feature set is also better compared to recognition performances using LCS and block-based features, individually.

The average recognition sensitivity without image rotation and with image rotation techniques using LCS, block-based and proposed combined features for Database I, Database II and Database III are shown in Figure 2.13. From Figure 2.13, it is seen that

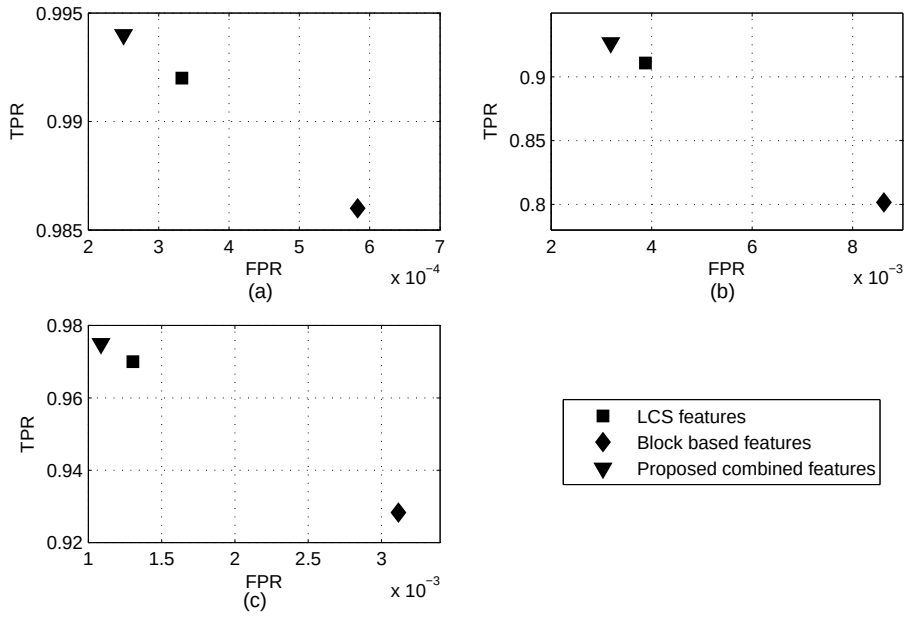


Figure 2.12: ROC graphs of hand gesture recognition with image rotation technique using LCS, block-based and proposed combined features for (a) Database I, (b) Database II and (c) Database III.

the proposed combined feature set provides better hand gesture recognition performance compared to LCS and block-based feature sets for all three hand gesture databases and both techniques (without and with image rotation). The proposed combined features provides better recognition performance compared to LCS and block-based features because it incorporates both contour as well as regional information of the hand gesture. On the other hand, LCS and block-based features incorporate only contour and regional information of hand gesture respectively. From Figure 2.13, it is also seen that the hand gesture recognition performance with image rotation technique is better compared to without image rotation technique. This indicates that the gesture recognition performance is improved due to proposed image rotation technique, where proposed image rotation technique makes segmented image rotation invariant using the direction of first principal component. Note that Databases II and III are constructed incorporating the variation of position, rotation and size of the gesture images under different illumina-

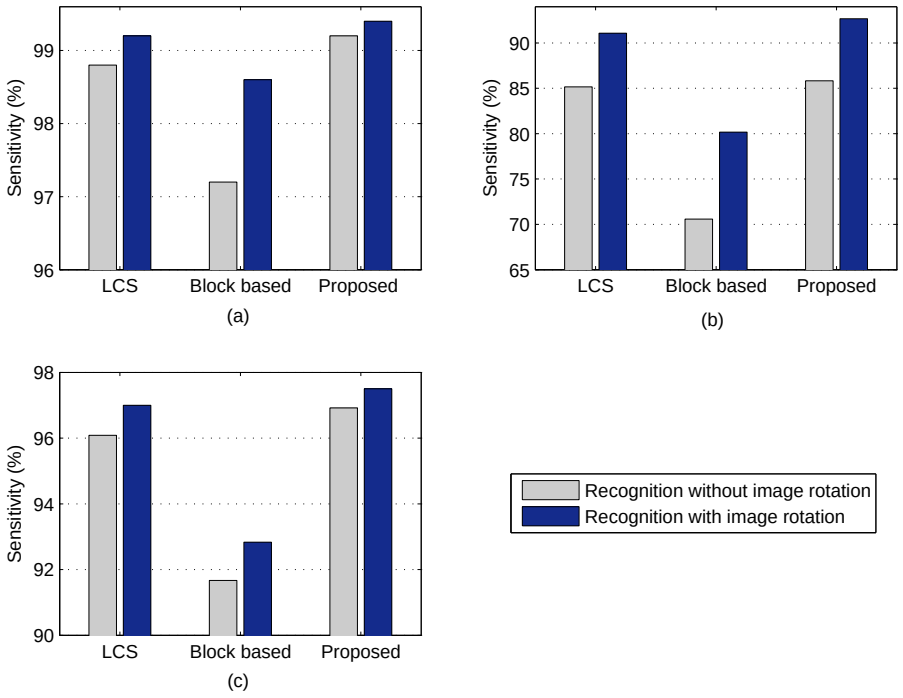


Figure 2.13: The average recognition sensitivity of without image rotation and with image rotation techniques using LCS, block-based and proposed combined features for (a) Database I, (b) Database II and (c) Database III.

tion conditions. Therefore, all the results provided in the above experimental study are obtained under the variation of position, rotation and size of the gesture images along with different illumination conditions.

2.5 Conclusions

In this chapter, homomorphic filtering and grayworld algorithm are used to normalize illumination variation of grayscale and color images respectively. Otsu and a skin color detection based segmentation algorithm are used to segment hand region from grayscale and color image and to provide binary silhouette. A novel image rotation technique is proposed in this chapter to make segmented image rotation invariant by coinciding the first principal component of the segmented hand gestures with vertical

axes. Morphological filtering technique is applied here to reduce object noise from binary image and to obtain a well defined smooth, closed, and complete segmented hand gesture image. A combined feature set is also proposed in this chapter to represent the static hand gesture image. The proposed combined feature set is obtained by appending LCS features with block-based features. A summary of the chapter is presented below:

- A novel image rotation technique is proposed to make segmented image as rotation invariant. In the proposed technique, the rotation normalization is achieved by coinciding the first principal component of the segmented hand gestures with vertical axes.

- A combined feature set by appending LCS features with block-based features is also proposed for better representation of static hand gesture image. This feature set is applied as input to the classifier.

- The proposed combined feature set is compared with LCS [16], block-based [17] feature sets using MLP-BP-NN as a classifier. Experiments are separately conducted on three different static hand gesture databases using LCS, block-based and proposed combined feature sets.

- Experimental results show that the proposed combined feature set provides better recognition performance compared to LCS [16] and block-based [17] feature sets individually. This is because the proposed combined feature set incorporates both contour as well as regional information of the hand gesture.

- To investigate the performance of the proposed image rotation technique, the experiments are separately conducted on three different static hand gesture databases using without image rotation technique and with image rotation technique. The experimental results demonstrate that gesture recognition performance has significantly improved due to proposed image rotation technique.

CHAPTER 3

Hand Gesture Recognition using DWT and F-ratio based Feature Set

3.1 Introduction

Localized contour sequence (LCS), block-based and proposed combined feature extraction techniques are described in detail in chapter 2. The proposed combined features provide higher recognition performance compared to LCS and block-based features to recognize the static gesture from its hand alphabet images. However, this combined feature set is unable to provide pixel-wise information of the gesture images. The pixel-wise information is important because it provides the local information and the occurrence of local variation of the gesture image [85]. Also, the statistical moments features [5,13,86] which are extracted from the binary silhouette of gesture image, do not incorporate the pixel-wise information. On the other hand, discrete cosine transform (DCT) coefficients [51] carry the overall pixel information of the hand gesture in the frequency domain and compact most of the energy of gesture image in the first few coefficients. However, DCT coefficients do not incorporate localized time-frequency information and also do not offer the multi-resolution analysis of an image. Therefore, a discrete wavelet transform (DWT) and Fisher ratio (F-ratio) based feature set is proposed in this chapter for better representation of hand gesture image. The proposed DWT and F-ratio based feature set offer more efficient representation of hand gesture image because proposed feature extraction technique uses DWT which has the following advantages: (i) it offers a simultaneous localization in time and spatial frequency domains; (ii) it provides multi-resolution analysis of an image using higher level decomposition; (iii) it is able to separate the fine details of an image; (iv) it is capable of revealing aspects of data where other signal analysis techniques may be missing the aspects like trends, breakdown points, discontinuities in higher derivatives and self-similarity; (v) it is invertible and can be orthogonal. In addition, the important DWT coefficient matrices are selected based on F-ratio based technique.

In the previous chapter MLP-BP-NN is used as a classifier to recognize the static gesture from hand alphabet images using combined feature sets. The MLP-BP-NN classifier achieves a significant recognition performance. However, MLP-BP-NN suffers

from local minima problems and it slowly converges to its local or global minima from randomly selected initial weights [87]. Apart from this, the classification performance of the MLP-BP-NN is highly affected by selection of the hidden nodes. There is no existing method that allows one for deciding the exact number of hidden nodes [88]. In this thesis, the number of hidden nodes is chosen empirically [80]. This chapter introduces another type of universal feed forward neural network known as radial basis function (RBF) neural network which was first formulated by Broomhead *et al.* [89]. The RBF neural network (RBF-NN) has become quite popular in pattern classification problem because it has the following advantages [2]: (i) its architecture is very simple, only one hidden layer consists between input and output layers; (ii) in hidden layer localized radial basis functions are used to perform nonlinear transform of input vector from input space to higher dimension hidden space where vector are more linearly separable; (iii) sensitivity of the hidden neuron is tuned by adjusting parameters of basis function; (iv) this network is faster and free from local minima problem.

This chapter proposes a feature extraction technique to extract DWT and F-ratio based (DWT-FR) feature set from their images and introduced RBF-NN as a classifier. In the proposed feature extraction technique, DWT is applied on the resized and enhanced grayscale image, and then the important DWT coefficient matrices are selected as features using the F-ratio based technique for representing the hand gesture image. The proposed DWT-FR feature set is applied as input to the RBF-NN to recognize the static gesture of hand alphabets. To verify gesture recognition performance of the proposed DWT-FR features and to compare it with the combined feature set as discussed in chapter 1 and other earlier reported techniques such as Krawtchouk moments (KM) features [5] and DCT features [51], experiments are separately carried out on three hand gesture databases using MLP-BP-NN and RBF-NN classifiers. Experimental results show that the proposed DWT-FR feature set provides better recognition performance compared to KM [5], DCT [51] and proposed combined feature sets for all three databases. Experimental results also demonstrate that RBF-NN classifier offers higher recognition performance compared to MLP-BP-NN classifier.

3.1.1 Organization of the Chapter

This chapter is organized as follows: Section 3.2 describes the basic theoretical background of wavelet transform (WT), Fisher ratio (F-ratio), Krawtchouk moments (KM), discrete cosine transform (DCT) and RBF neural network. The proposed framework is described in Section 3.3. Recognition performances are evaluated in Section 3.4. Finally, the chapter concludes with some remarks in Section 3.5.

3.2 Theoretical Background

3.2.1 Wavelet Transform (WT)

The wavelet transform (WT) [90] is one of the most important and powerful tools which can be used for one or multi-dimensional data analysis instead of classical Fourier transform (FT). WT was first introduced by Morlet, who used it to analyze seismic signal [91]. It has wide range of applications in the area of physics, signal processing, digital image processing, speech processing, computer graphics and video [90]. In the classical FT, the signal is decomposed based on complex exponentials, whereas the WT decomposes the signal over a set of dilated and translated wavelets. Though the short time Fourier transform (STFT) decomposes the signal using a constant window size and provides better time resolution than FT. However, it has a fixed resolution in time and frequency, and it processes all frequencies in the same way. On the other hand, WT executes a multiresolution analysis, means it processes different frequencies in different ways [92]. This technique provides a good frequency localization when analyzing the low-frequency part of the signal and a good time localization when analyzing the high frequency portion of the same signal. Therefore, WT is suited for most signal and image processing applications [92]. The theoretical aspects of wavelet transforms: (i) continuous wavelet transform (CWT) and (ii) discrete wavelet transform (DWT) are described as follows.

3.2.1.1 Continuous Wavelet Transform (CWT)

Let $f(x)$, be a finite energy signal. The CWT of a signal $f(x) \in L^2(R)$ can be defined [92] as

$$W_\psi(s,\tau) = \frac{1}{\sqrt{s}} \int_{-\infty}^{+\infty} f(x)\psi^*\left(\frac{x-\tau}{s}\right) dx \tag{3.1}$$

where $\psi(x)$ is mother wavelet or wavelet kernel, s and τ are continuous scaling and shift parameters respectively. The $\psi^*(x)$ represents complex conjugate of $\psi(x)$. A wavelet transform is invertible if the mother wavelet satisfies the admissibility condition, that is represented by (3.2) [93].

$$C_\psi = \int_0^\infty \frac{\left|\widehat{\psi}(\omega)\right|^2}{|\omega|} d\omega < \infty \tag{3.2}$$

where, $\widehat{\psi}(\omega)$ represents the FT of $\psi^*(x)$. In most of the cases, admissible condition implies zero-mean mother wavelet function as $\psi(0) = \int_{-\infty}^{\infty} \psi(x)dx = 0$. Thus, for a given $W_f(s,\tau)$, $f(x)$ can be obtain by the inverse CWT and this is represented by (3.3) [94].

$$f(x) = \frac{1}{C_\psi} \int_0^\infty \int_{-\infty}^\infty W_\psi(s,\tau) \frac{1}{\sqrt{s}} \psi\left(\frac{x-\tau}{s}\right) \frac{d\tau ds}{s^2} \tag{3.3}$$

At every possible scale and translation, calculating wavelet coefficients for a data is computationally expensive and not practical as well.

3.2.1.2 Discrete Wavelet Transform (DWT)

The continuous scaling and shift parameters are discretized using two integer indexes (m,n) to reduce the redundancy and to make the wavelet transform more practical. Let, the scaling and shift parameters are defined as $s = s_0^m$ and $\tau = n\tau_0 s_0^m$, respectively, where, $s_0 > 1$ and $\tau_0 \neq 0$ represent discrete scale and shift steps, respectively. The dyadic scales and positions (where $s_0 = 2$ and τ_0=1) are selected to analyze a signal

more efficiently. The DWT with dyadic scales and positions can be defined [95] as

$$W_\psi(m,n) = \frac{1}{\sqrt{2^m}} \int_{-\infty}^{+\infty} f(x)\psi\left(\frac{x-2^m n}{2^m}\right) dx,\ \ m,n \in Z \tag{3.4}$$

which corresponds to $s = s_0^m = 2^m$ and $\tau = n\tau_0 s_0^m = n2^m$. Note that the translation step is dependent on the dilation, therefore short wavelets are advanced by small steps, and long ones by large steps [96]. The function $f(x) \in L^2(R)$ can be expressed as

$$f(x) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} W_\psi(m,n)\,\psi_{m,n}(x) \tag{3.5}$$

where

$$\psi_{m,n}(x) = \frac{1}{\sqrt{2^m}}\psi\left(\frac{x-2^m n}{2^m}\right) = 2^{-m/2}\psi\left(\frac{x}{2^m} - n\right) \tag{3.6}$$

The wavelet functions $\psi_{m,n}(x)$ are orthonormal in both indexes if it satisfy the following condition [97]

$$\int_{-\infty}^{+\infty} \psi_{m,n}(x)\,\psi_{p,q}(x)dx = \begin{cases} 1, & m = p, n = q \\ 0, & otherwise \end{cases} \tag{3.7}$$

The scaling functions of multiresolution analysis which are used as complementary basis of DWT, also satisfy the following orthonormal condition within the same scale [95].

$$\int_{-\infty}^{+\infty} \phi_{m,n}(x)\,\phi_{m,q}(x)dx = \delta_{n-q} \tag{3.8}$$

where $\phi_{m,n}(x)$ is scaling function set and it can be defined as

$$\phi_{m,n}(x) = 2^{-m/2}\phi\left(\frac{x}{2^m} - n\right). \tag{3.9}$$

Moreover, the scaling coefficients of $f(x)$ are defined as

$$W_\phi(m,n) = \frac{1}{\sqrt{2^m}} \int_{-\infty}^{+\infty} f(x)\,\phi\left(\frac{x-2^m n}{2^m}\right) dx. \tag{3.10}$$

The wavelet and scaling bases jointly satisfy their complementary basis property of orthonormal and it can be expressed [95] as

$$\int_{-\infty}^{+\infty} \psi_{m,n}(x)\phi_{p,q}(x)dx = 0 \quad for\ \forall\, m, n, p, q. \tag{3.11}$$

Note that the function $f(x)$ can be expressed only in terms of discrete wavelet functions as shown in (3.5), using infinite number of scales. However, if the scaling and wavelet basis functions are jointly utilized, the function $f(x)$ can be expressed by finite number of resolutions as follows

$$f(x) = \sum_{n=-\infty}^{\infty} W_\phi(m_0, n)\phi_{m_0,n}(x) + \sum_{m=m_0}^{\infty}\sum_{n=-\infty}^{\infty} W_\psi(m, n)\,\psi_{m,n}(x) \tag{3.12}$$

where, the first term represents a low-pass approximation to $f(x)$ utilizing the scaling functions at scale m_0 and latter term expresses wavelet representation of the detail signal [95]. This representation of $f(x)$ in (3.12) is more efficient than representation in (3.5) and also highlights the significant role of the scaling basis in wavelet transforms based multiresolution signal decomposition framework. Mallat has developed an efficient way to implement this scheme using FIR filter banks [98]. DWT can be realized by passing the signal through a series of quadrature mirror filters with rescaling. The resolution of the signal and scale are determined by filtering and sub sampling operations respectively. The procedure of DWT-based multiresolution decomposition of an one dimensional (1D) signal, $f(x)$ is shown in Figure 3.1. Two digital filters and two down samplers that down sample the filtered signal by 2, are used at each stage of this scheme [63]. The first filter, G_0 which is high-pass in nature, used as the discrete mother wavelet, and the second, H_0 which is low-pass in nature, is the mirror version of the first. The down sampled outputs corresponding to high-pass and low-pass filter provides detail coefficients (D1) and approximation coefficients (A1), respectively. The first approximation coefficients, A1 is further decomposed and this process continues.

The relation of the impulse response of filter G_0 and filter H_0 is represented by [98]

$$g(k) = (-1)^{1-k} h(k). \tag{3.13}$$

In case of two-dimensional (2D) signal $f(x, y)$, like image, 2D DWT can be implemented

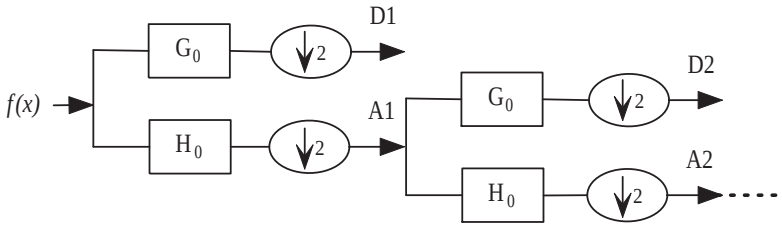


Figure 3.1: DWT-based subband decomposition for an 1D signal $f(x)$.

using digital filters and down samplers. In this way, at each level of decomposition, the $f(x, y)$ can be decomposed in specific sets of coefficients [67]. Figure 3.2 shows the subband decomposition using 2D-DWT for an image. In the figure, the low pass and high pass filters are denoted by H_0 and G_0 respectively. The image is first filtered along row-wise and the filtered images are down sampled its columns by a factor of 2 without losing any information. After down sampling, filtered images are again filtered along column-wise and down sampled its rows by a factor of 2. Therefore, four sub-images are obtained at decomposition level 1, and sub-images are represented as $A1$, $D1^H$, $D1^V$ and $D1^D$, where $A1$ sub-image carries approximation coefficients which contain the low frequency information in both horizontal and vertical directions, $D1^H$ sub-image carries horizontal detail coefficients which contain the high and low frequency information in horizontal and vertical directions respectively, $D1^V$ sub-image carries vertical detail coefficients which contain the low and high frequency information in horizontal and vertical directions respectively and $D1^D$ sub-image carries diagonal detail coefficients which contain the high frequency information in both horizontal and vertical directions. The sub-image, $A1$ is further decomposed and this process continues.

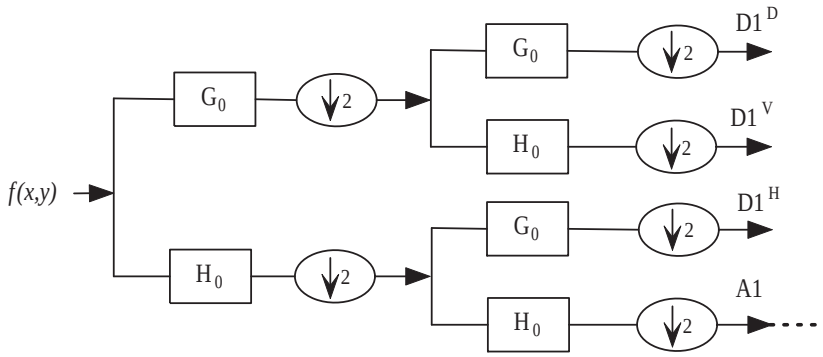


Figure 3.2: Subband decomposition using 2D-DWT for an image represented $f(x, y)$.

3.2.2 Fisher Ratio (F-ratio)

F-ratio [99] is a statistical measure for the analysis of variance of multi-class data. It is defined as

$$F - ratio = \frac{Variance\,of\,means\,between\,the\,classes}{Average\,Variance\,within\,the\,classes} \tag{3.14}$$

If k numbers of class are available, n_j is the total number of data points for j^{th} class, μ_j represents mean of j^{th} class, $\overline{\mu}$ is the total mean, x_{ij} is i^{th} data of j^{th} class then (3.14) can be represented [100] as

$$F - ratio = \frac{\frac{1}{k}\sum_{j=1}^{k}(\mu_j - \overline{\mu})^2}{\frac{1}{k}\sum_{j=1}^{k}\frac{1}{n_j}\sum_{i=1}^{n_j}(x_{ij} - \mu_j)^2} \tag{3.15}$$

where

$$\mu_j = \frac{1}{n_j}\sum_{i=1}^{n_j} x_{ij}$$

and

$$\bar{\mu} = \frac{1}{k}\sum_{j=1}^{k} \mu_j.$$

The value of $F - ratio$ increases on increasing the inter-class variance or decreasing the intra-class variance.

3.2.3 Krawtchouk Moments (KM)

The Krawtchouk moments (KM) [101] are a set of discreet orthogonal moments derived from Krawtchouk polynomials. The classical Krawtchouk polynomial of order n [101, 102] is represented as

$$\begin{aligned} K_n\left(x;p,N\right) &= {}_2F_1\left(-n,-x;-N;\tfrac{1}{p}\right) \\ x,n &= 0,1,2,3,\cdots,N;\;\; N>0\,;\; p\in(0,1) \end{aligned} \tag{3.16}$$

where N and p are the number of spatial points and the parameter of the binomial distribution, respectively. The ${}_2F_1$ is the hyper geometric function and it can be defined as

$${}_2F_1\left(a,b;c;z\right) = \sum_{m=0}^{\infty} \frac{(a)_m (b)_m z^m}{(c)_m m!}, \tag{3.17}$$

where, $(a)_m$ is the Pochammer symbol [101], represented as

$$(a)_m = a\left(a+1\right)\cdots\cdots\left(a+m-1\right) = \frac{\Gamma(a+m)}{\Gamma(a)}. \tag{3.18}$$

A set of $(N+1)$ Krawtchouk polynomials $\{K_n\left(x;p,N\right)\}$ forms a complete set of discrete basis functions [101] with weight function

$$W\left(x;p,N\right) = \frac{N!}{x!\left(N-x\right)!} p^x (1-p)^{N-x} \tag{3.19}$$

and meets the orthogonality condition

$$\begin{aligned} &\sum_{x=0}^{N} W\left(x;p,N\right) K_n\left(x;p,N\right) K_m\left(x;p,N\right) = \rho\left(n;p,N\right)\delta[n-m], \\ &\qquad n,m = 0,1,2,\cdots\cdots,N \end{aligned} \tag{3.20}$$

where $\delta[.]$ is the Kronecker delta function [5] and $\rho\left(n;p,N\right) = (-1)^n \left(\frac{1-p}{p}\right)^n \frac{n!}{(-N)_n}$. The set of weighted Krawtchouk polynomials $\left\{\bar{K}_n\left(x;p,N\right)\right\}$ for $n = 0,1,2,\cdots,N$ is defined [5] as

$$\bar{K}_n\left(x;p,N\right) = K_n\left(x;p,N\right)\sqrt{\frac{W\left(x;p,N\right)}{\rho\left(n;p,N\right)}} \tag{3.21}$$

Therefore, the orthogonality condition in (3.20) becomes

$$\sum_{x=0}^{N} \bar{K}_n\left(x; p, N\right) \bar{K}_m\left(x; p, N\right) = \delta[n-m]. \tag{3.22}$$

and an orthonormal basis is formed by the weighted Krawtchouk polynomials. The constant p works as a translation parameter that shifts the support of the polynomials [5]. If p deviates from 0.5 to $(0.5 + \Delta p)$, the support of weighted Krawtchouk polynomials are approximately shifted by $N\Delta p$ [101]. The sign of Δp represents the direction of shifting, when Δp is positive, the weighted Krawtchouk polynomials are shifted in $+x$ direction and vice versa [101].

The Kawtchouk moments of order n for N point one-dimensional signal $f(x)$ is defined [102] as

$$\begin{aligned} Q_n &= \sum_{x=0}^{N-1} \bar{K}_n\left(x; p, N-1\right) f(x), \\ p &\in (0,1)\,, \quad x, n = 0, 1, 2, \cdots, N-1. \end{aligned} \tag{3.23}$$

The inverse transform can be computed using

$$\begin{aligned} \hat{f}(x) &= \sum_{n=0}^{N-1} Q_n \bar{K}_n\left(x; p, N-1\right)\,, \\ x, n &= 0, 1, 2, \cdots, N. \end{aligned} \tag{3.24}$$

The Krawtchouk moments of order $(n+m)$ in terms of weighted Krawtchouk polynomials, for a 2D intensity function of image with size $N \times M$, $f(x,y)$, is defined [101] as

$$\begin{aligned} Q_{n,m} &= \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \bar{K}_n\left(x; p_1, N-1\right) \bar{K}_m\left(x; p_2, M-1\right) f(x,y), \quad p_1, p_2 \in (0,1)\,, \\ x, n &= 0, 1, 2, \cdots, N-1, \quad y, m = 0, 1, 2, \cdots, M-1 \end{aligned} \tag{3.25}$$

Using orthonormal property, the inverse transform is represented [5] as

$$\begin{aligned} \hat{f}(x,y) &= \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} Q_{n,m}\, \bar{K}_n\left(x; p_1, N\right) \bar{K}_m\left(x; p_2, M\right)\,, \\ x, n &= 0, 1, 2, \cdots, N-1, \quad y, m = 0, 1, 2, \cdots, M-1. \end{aligned} \tag{3.26}$$

3.2.4 Discrete Cosine Transform (DCT)

A discrete cosine transform (DCT) [103] provides a way to break a finite sequence of data points and expressed by a sum of cosine functions oscillating at different frequencies. DCT has numerous applications in science and engineering, mainly used for lossy compression of audio (MPEG) and images (JPEG) [104]. Similar to other transforms, DCT efforts to decorrelate the data and each transform coefficient can be encoded independently by keeping sufficient compression efficiency. The DCT of a 1D data sequence $f(x)$, $x = 0, 1, 2, \cdots (M-1)$ can be defined [104] as

$$G(u) = \sum_{x=0}^{M-1} \alpha(u) f(x) \cos\left(\frac{(2x+1)u\Pi}{2M}\right) \; for \; u = 0, 1, 2, \cdots (M-1), \tag{3.27}$$

where $G(u)$ indicates the u^{th} DCT coefficient. Similarly, the inverse discrete cosine transform (IDCT) is defined as

$$f(x) = \sum_{u=0}^{M-1} \alpha(u) G(u) \cos\left(\frac{(2x+1)u\Pi}{2M}\right) \; for \; x = 0, 1, 2, \cdots (M-1). \tag{3.28}$$

For both of the cases as in (3.27) and (3.28), $\alpha(u)$ defined as

$$\alpha(u) = \begin{cases} \frac{1}{\sqrt{M}}, & for \; u = 0, \\ \sqrt{\frac{2}{M}}, & for \; u \neq 0. \end{cases} \tag{3.29}$$

From (3.27) and (3.29), it is seen that $G(0) = \frac{1}{\sqrt{M}} \sum_{x=0}^{M-1} f(x)$. Therefore, first DCT coefficient $G(0)$ represents the average value of the sample sequences. The first DCT coefficient is referred to as the DC Coefficient and all other coefficients are known as AC Coefficients [105].

Assume, a 2D sequence represented as $f(x, y)$, $x, y = 0, 1, 2, \cdots (M-1)$, the DCT of

this sequence 2D-DCT is the direct extension of 1D-DCT, can be defined [105] as

$$G(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{M-1} \alpha(u)\alpha(v) f(x,y) \cos\left(\frac{(2x+1)u\Pi}{2M}\right) \cos\left(\frac{(2y+1)v\Pi}{2M}\right) \quad for\ u,v = 0,1,2,\cdots(M-1). \tag{3.30}$$

Similarly, 2D IDCT can be defined as

$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{M-1} \alpha(u)\alpha(v) G(u,v) \cos\left(\frac{(2x+1)u\Pi}{2M}\right) \cos\left(\frac{(2y+1)v\Pi}{2M}\right) \quad for\ x,y = 0,1,2,\cdots(M-1). \tag{3.31}$$

The $\alpha(u)$ and $\alpha(v)$ are defined in (3.29).

3.2.5 Radial Basis Function (RBF) Neural Network

The radial basis function (RBF) neural network [80] is widely used in pattern recognition tasks for faster learning property and compact topology than other networks. The RBF neural network (RBF-NN) is a feed-forward neural network which consists of one hidden layer with RBF units between input and output layer. The hidden layer is primarily responsible for computation and the output layer provides the output of the network corresponding to the input patterns. The basic structure of this network is shown in Figure 3.3 [80]. In this network, the input vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ is projected into hidden space in a nonlinear manner using RBF function and hidden space to output space in a linear manner by multiplying weight vector with it.

The output of the i^{th} RBF unit in hidden layer is represented as

$$G_i(\mathbf{x}) = G_i\left(\|\mathbf{x} - \mathbf{c}_i\|\right), \quad i = 1,2,3,\ldots,m \tag{3.32}$$

where, $\mathbf{x}$ is an n-dimensional input vector, $\mathbf{c}_i$ is a center vector of RBF unit with the same dimension as $\mathbf{x}$, m is the number of hidden units and $G_i(.)$ is the i^{th} RBF unit.

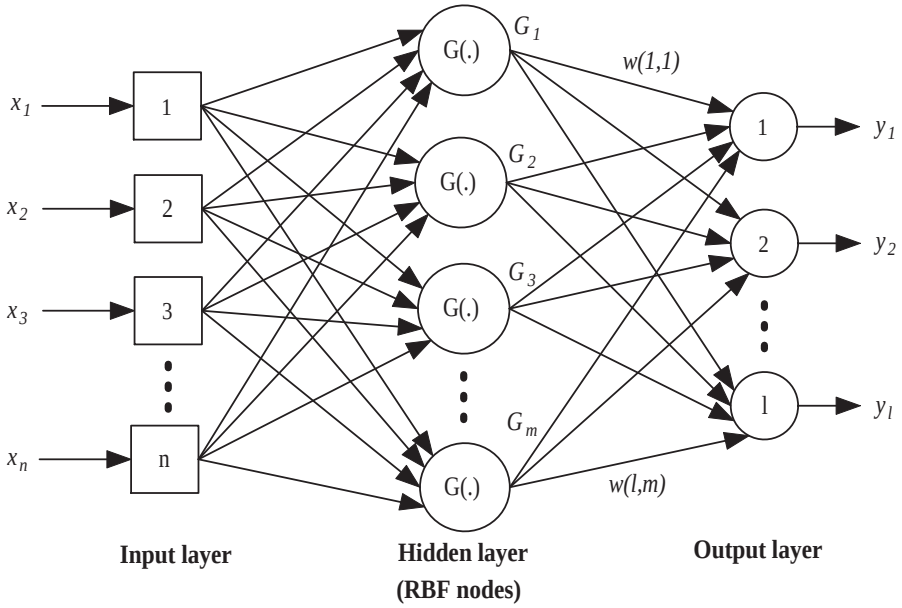


Figure 3.3: Structure of radial basis function neural network.

Typically, $G_i(.)$ is chosen as a Gaussian function

$$G_i(\mathbf{x}) = G_i\left(-\frac{\|\mathbf{x}-\mathbf{c}_i\|}{\sigma^2}\right), \tag{3.33}$$

where, σ is the spread factor or width of Gaussian function. The j^{th} output $y_j(\mathbf{x})$ of RBF neural networks is represented as

$$y_j(\mathbf{x}) \;=\; w(j,0) \;+\; \sum_{i=1}^{m} w(j,i)\; G_i(\mathbf{x}) \tag{3.34}$$

where, $w(j,i)$ is the weight of the i^{th} receptive field to the j^{th} output and $w(j,0)$ is the bias of the j^{th} output. In the following analysis, the bias is not considered in order to reduce the network complexity. Hence, the j^{th} output $y_j(\mathbf{x})$ of an RBF neural network is represented as

$$y_j(\mathbf{x}) \;=\; \sum_{i=1}^{m} w(j,i)\; G_i(\mathbf{x}) \tag{3.35}$$

An RBF neural network classifier, the actual target value corresponding to j^{th} output

node for identifying class membership of $\mathbf{x}$ [106] is defined as

$$t_j(\mathbf{x}) = y_j(\mathbf{x}) + e_j(\mathbf{x}) = \sum_{i=1}^{m} w(j,i)\ G_i(\mathbf{x}) + e_j(\mathbf{x}) \tag{3.36}$$

For all output node (3.36) is defined as

$$\mathbf{t} = \mathbf{wG} + \mathbf{e} \tag{3.37}$$

where, $\mathbf{t} = [t_1, t_2, \ldots, t_l]^T$ is the target vector, $\mathbf{e} = [e_1, e_2, \ldots, e_l]^T$ is the error vector and $\mathbf{G} = [G_1, G_2, \ldots, G_m]^T$ is the projected input vector in hidden space and

$$\mathbf{w} = \begin{bmatrix} w(1,1) & w(1,2) & w(1,3) & \cdots & w(1,m) \\ w(2,1) & w(2,2) & w(2,3) & \cdots & w(2,m) \\ \vdots & & & & \\ w(l,1) & w(l,2) & w(l,3) & \cdots & w(l,m) \end{bmatrix}$$

is weight matrix. The optimal weights are estimated by solving (3.37) using the pseudo inverse method [80]. Therefore, the estimated weights are

$$\mathbf{w}^* = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{t} \tag{3.38}$$

where $\mathbf{G}^T$ means the transpose of $\mathbf{G}$ and $(\mathbf{G}^T\mathbf{G})^{-1}$ represents the inverse matrix of $\mathbf{G}^T\mathbf{G}$. There are also other methods, like the Fisher method [107] or the cross-entropy method [108] which have used for estimating the weights for RBF neural network.

3.3 Proposed Framework: DWT and F-ratio based Feature Extraction Technique

Features are used to represent a pattern with the goal of minimizing the loss of important information. This chapter proposes a feature extraction technique to extract DWT and F-ratio based feature set for representation of static hand gesture image. The

proposed technique includes the following steps.

3.3.1 Cropping and Resizing the Image

In this step, a bounding box is constructed around the hand region of rotation invariant segmented image and using this bounding box, hand region is cropped from its corresponding rotation invariant enhanced grayscale image. The cropped image is then resized to 64×64 for Database I and 128×128 for Databases II and III using nearest-neighbor interpolation method. Note that, for complex background cases, cropped hand region of rotation invariant enhanced grayscale image is masked using cropped filled image of rotation invariant segmented image. Figs. 3.4 (a), (b), (c) and (d) show rotation invariant segmented image, rotation invariant enhanced image, cropped hand region of rotation invariant enhanced image and resized cropped hand region image respectively for a grayscale hand gesture image of Database II. Figs. 3.5 (a), (b), (c), (d) and (e) present rotation invariant segmented image, rotation invariant enhanced image, cropped filled image of rotation invariant segmented image, cropped hand region of rotation invariant enhanced grayscale image after masking using cropped filled image and resized cropped hand region image respectively for a color hand gesture image of Database III.

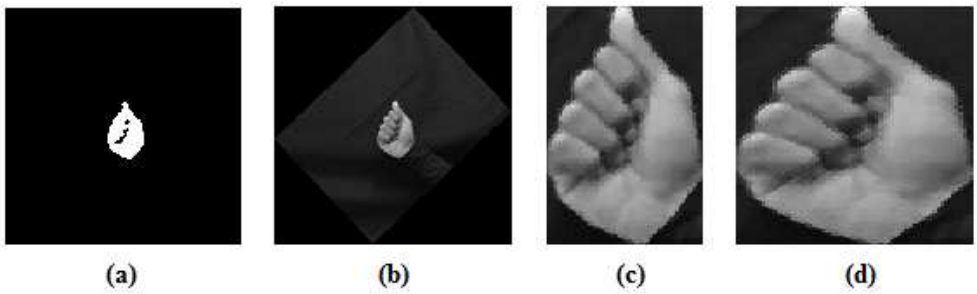


Figure 3.4: Results of following steps for a grayscale hand gesture image of Database II. (a) Rotation invariant segmented image. (b) Rotation invariant enhanced image. (c) Cropped hand region of rotation invariant enhanced image. (d) Resized cropped hand region image.

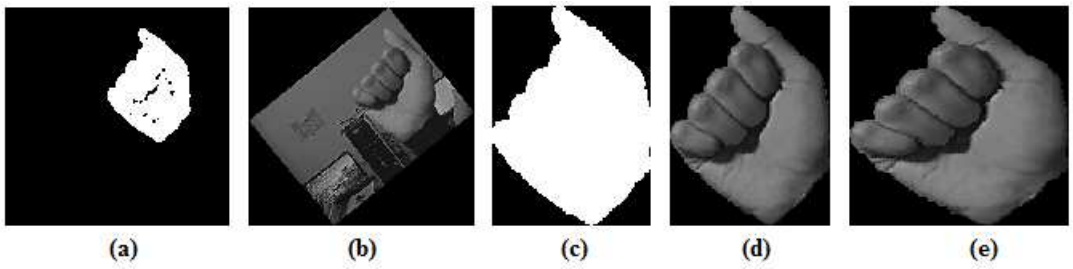


Figure 3.5: Results of following steps for a color hand gesture image of Database III. (a) Rotation invariant segmented image. (b) Rotation invariant enhanced grayscale image. (c) Cropped filled image of rotation invariant segmented image. (d) Cropped hand region of rotation invariant enhanced grayscale image after masking using cropped filled image. (e) Resized cropped hand region image.

3.3.2 Applying DWT on Resized and Enhanced Grayscale Image

According to the procedure of multiresolution decomposition [98] as described in Section 3.2.1.2, the image is decomposed in four coefficient matrices (approximation, horizontal detail, vertical detail and diagonal detail) at each level of decomposition [67]. For higher levels decomposition, subsequent digital filtering and down sampling operations are involved on approximation coefficient matrices. The wavelet decomposition tree upto m level is shown in Figure 3.6, where CA_i, CH_i, CV_i and CD_i represent

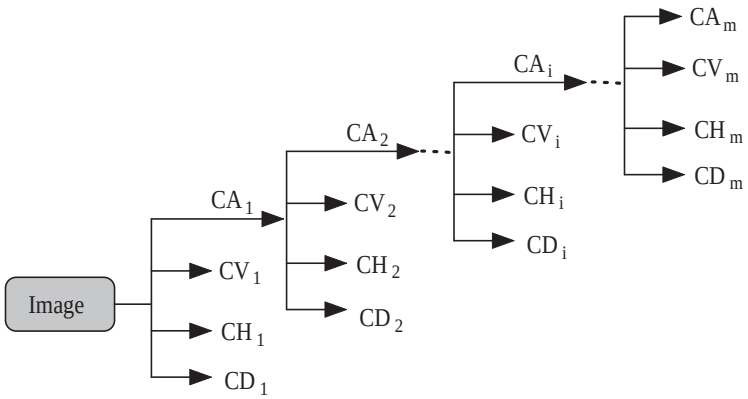


Figure 3.6: Two-dimensional wavelet decomposition tree upto m level.

approximation, horizontal detail, vertical detail and diagonal detail coefficient matrices

respectively at i^{th} decomposition level. In this work, Haar wavelet transform is applied on resized grayscale image to extract the approximation and detail coefficients. In Haar wavelet [67], the wavelet and scaling functions are separately defined by (3.39) and (3.40).

$$\psi(x) = \begin{cases} 1 & 0.0 \leq x < 0.5 \\ -1 & 0.5 \leq x \leq 1.0 \\ 0 & otherwise \end{cases} \tag{3.39}$$

$$\phi(x) = \begin{cases} 1 & 0.0 \leq x < 1.0 \\ 0 & otherwise \end{cases} \tag{3.40}$$

The Haar wavelet is used here because: (i)) it is simple, orthogonal and computationally efficient [109], (ii) it uses just two scaling and wavelet function coefficients and calculates pair wise averages and differences, (iii) it does not have overlapping windows, and reflects only changes between adjacent pixel pairs [110] and (iv) it effectively describes the soft parts of human body [111]. The levels of decomposition is selected based on the size of the resized image. In this work, 5 levels of decomposition is chosen for the images of Database I (where resized image size is $2^6 \times 2^6$) and 6 levels of decomposition is chosen for the images of Databases II and III (where resized image size is $2^7 \times 2^7$).

3.3.3 Selection of DWT Coefficient Matrices using F-ratio

Selection of DWT coefficient matrices is very important for feature representation of a gesture image. If all coefficient matrices are used to represent the gesture, the dimension of the feature vector will be very high and therefore, the system will be more computationally complex. A large dimensional feature vector also reduces the classification performance due to the redundant or irrelevant coefficient matrices. To solve this problem, this work proposes an F-ratio based coefficient matrices selection technique to select the important coefficient matrices for each gesture. The proposed F-ratio based coefficient matrices selection technique involves the following steps.

1. Initially, calculate $F - ratio$ value for DWT coefficients by (3.15).

2. Compute average $F-ratio$ value of each DWT coefficients matrix.

3. Find the maximum of average $F-ratio$ values for all coefficient matrices.

Table 3.1: The number of coefficients and average F-ratio values of the DWT coefficient matrices for Databases I, II and III

Database I		Databases II and III		Average F-ratio values		
Coefficient matrices	Number of coefficients	Coefficient matrices	Number of coefficients	Database I	Database II	Database III
–	–	CA6	4	–	0.75	1.93
–	–	CH6	4	–	0.42	0.92
–	–	CV6	4	–	0.74	1.21
CA5	4	CD6	4	7.32	**1.17**	**2.27**
CH5	4	CH5	16	**9.49**	0.63	1.16
CV5	4	CV5	16	6.95	0.50	1.09
CD5	4	CD5	16	8.59	0.39	0.67
CH4	16	CH4	64	6.24	0.32	0.59
CV4	16	CV4	64	5.93	0.34	0.63
CD4	16	CD4	64	2.72	0.14	0.24
CH3	64	CH3	256	3.34	0.16	0.26
CV3	64	CV3	256	3.43	0.19	0.32
CD3	64	CD3	256	1.50	0.05	0.07
CH2	256	CH2	1024	1.85	0.09	0.12
CV2	256	CV2	1024	2.05	0.10	0.13
CD2	256	CD2	1024	0.48	0.02	0.02
CH1	1024	CH1	4096	0.71	0.04	0.05
CV1	1024	CV1	4096	0.77	0.04	0.05
CD1	1024	CD1	4096	0.08	0.01	0.01

4. Compute the threshold value Th as

$$Th = \frac{1}{L-1}(\text{Maximum average F} - \text{ratio value}), \tag{3.41}$$

where L is the number of decomposition levels.

5. Select the coefficient matrices having average $F-ratio$ value more than Th.

The number of coefficients and average F-ratio values of the DWT coefficient matrices for Databases I, II and III are shown in Table 3.1.

From Table 3.1, it is observed that the maximum average $F-ratio$ value of DWT coefficient matrices for Databases I, II and III are 9.49, 1.17 and 2.27 respectively. Therefore, based on the proposed coefficient matrices selection technique, the threshold (Th) values are selected as 2.37, 0.23 and 0.45 for Databases I, II and III respectively. From Table 3.1, it is also observed that according to Th values, $CA5$, $CH5$, $CV5$, $CD5$, $CH4$, $CV4$, $CD4$, $CH3$ and $CV3$ coefficient matrices are selected for Database I and $CA6$, $CH6$, $CV6$, $CD6$, $CH5$, $CV5$, $CD5$, $CH4$ and $CV4$ coefficient matrices are selected for Databases II and III using proposed F-ratio-based coefficient matrices selection technique to represent each hand gesture image. In this way, only 192 DWT coefficients are used as extracted features to represent each 64×64 (or 4,096) pixels image of Database I and 128×128 (or 16,384) pixels image of Databases II and III.

3.4 Performance Evaluation

The performance of gesture recognition using the proposed feature extraction method is evaluated on three static hand gesture databases in the similar way as described in chapter 2. In this chapter, the gestures are classified based on two classifiers: MLP-BP-NN and RBF-NN. To verify gesture recognition performance of the proposed DWT and F-ratio based (DWT-FR) features and to compare it with the combined features which is shown in chapter 1 and other earlier reported techniques such as Kawtchouk moments (KM) [5] and discrete cosine transform (DCT) features [51], the experiments are separately conducted using MLP-BP-NN and RBF-NN classifier.

3.4.1 Experimental Results using MLP-BP-NN Classifier

The experiments are conducted on three hand alphabets databases of static gesture. First one is the database of 25 static gesture of D/ISL hand alphabets, and second and third are the databases of 24 static gesture of ASL hand alphabets. The training and testing datasets are constructed in a similar manner as discussed in chapter 2. Randomly selected 50% data of each database are used to train the classifier and the

remaining 50% data are used to evaluate the recognition performance. The recognition performance is evaluated using four common metrics including accuracy (*Acc*), sensitivity (*Sen*), positive predictivity (*Ppr*) and specificity (*Spe*) [82]. The common metrics are clearly described in chapter 2. To analyze and compare the hand gesture recognition performance, the experiments are separately conducted on three databases using KM, DCT, combined and DWT-FR features with MLP-BP-NN classifier. The comparative results of gesture recognition performance using KM [5], DCT [51], proposed combined and DWT-FR features with MLP-BP-NN classifier are tabulated in Table 3.2. The

Table 3.2: Comparative performance of hand gesture recognition using KM, DCT, combined and DWT-FR feature sets with MLP-BP-NN classifier for three distinct databases

Database	Feature set	Acc (%)	Sen (%)	Ppr (%)	Spe (%)
Database I	KM [5]	99.92	99.00	99.05	99.96
	DCT [51]	99.97	99.60	99.62	99.98
	Combined	99.95	99.40	99.43	99.98
	DWT-FR	**99.98**	**99.80**	**99.81**	**99.99**
Database II	KM [5]	99.06	88.67	89.09	99.51
	DCT [51]	99.26	91.17	91.33	99.62
	Combined	99.39	92.67	92.75	99.68
	DWT-FR	**99.53**	**94.33**	**94.49**	**99.75**
Database III	KM [5]	99.60	95.17	95.61	99.79
	DCT [51]	99.78	97.42	97.55	99.89
	Combined	99.79	97.50	97.55	99.89
	DWT-FR	**99.90**	**98.75**	**98.78**	**99.95**

comparative results indicate that the average accuracy, sensitivity, positive predictivity and specificity of hand gesture recognition with MLP-BP-NN classifier using KM features are 99.92%, 99.00%, 99.05% and 99.96% respectively for Database I, 99.06%, 88.67%, 89.09% and 99.51% respectively for Database II and 99.60%, 95.17%, 95.61% and 99.79% respectively for Database III, using DCT features are 99.97%, 99.60%, 99.62% and 99.98% respectively for Database I, 99.26%, 91.17%, 91.33% and 99.62%

respectively for Database II and 99.78%, 97.42%, 97.55% and 99.89% respectively for Database III, and using combined features are 99.95%, 99.40%, 99.43% and 99.98% respectively for Database I, 99.39%, 92.67%, 92.75% and 99.68% respectively for Database II and 99.79%, 97.50%, 97.55% and 99.89% respectively for Database III. On the other hand, the average accuracy, sensitivity, positive predictivity and specificity of hand gesture recognition using DWT-FR features with MLP-BP-NN classifier are 99.98%, 99.80%, 99.81% and 99.99% respectively for Database I, 99.53%, 94.33%, 94.49% and 99.75% respectively for Database II and 99.90%, 98.75%, 98.78% and 99.95% respectively for Database III. The comparative recognition performance is also analyzed in terms of ROC graph. The ROC graphs of hand gesture recognition using KM features with MLP-BP-NN classifier, DCT features with MLP-BP-NN classifier, combined features with MLP-BP-NN classifier and DWT-FR features with MLP-BP-NN classifier for three distinct databases are shown in Figure 3.7. The Table 3.2 and Figure 3.7 show

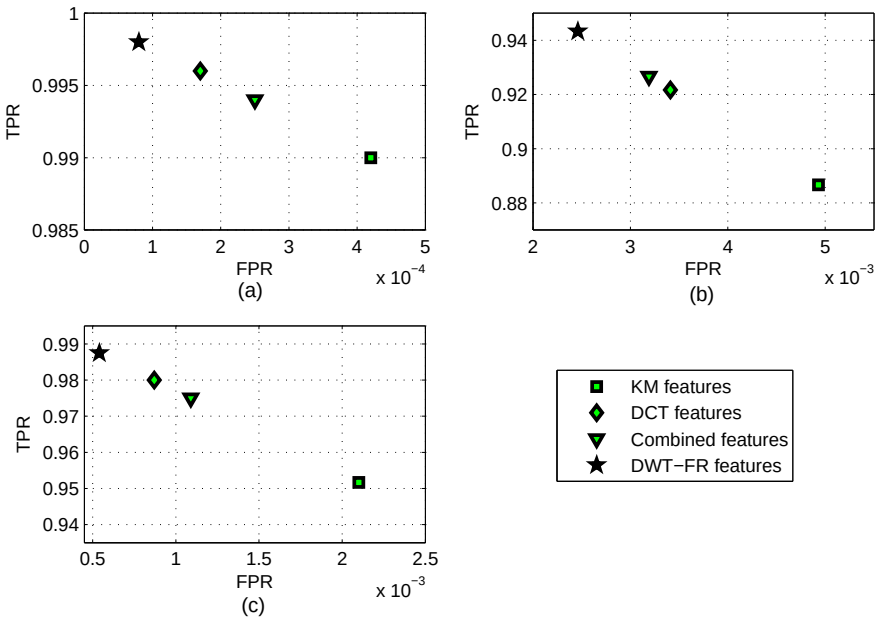


Figure 3.7: ROC graphs of hand gesture recognition using KM, DCT, combined and DWT-FR features with MLP-BP-NN classifier for (a) Database I, (b) Database II and (c) Database III.

that using MLP-BP-NN classifier, the DWT-FR features provides better recognition performance compared to KM [5], DCT [51], and combined feature sets for all three databases.

3.4.2 Experimental Results using RBF-NN Classifier

To evaluate the hand gesture recognition performance, the experiments are also separately conducted on three databases using the KM, DCT, combined and DWT-FR features with RBF-NN classifier. The comparative results of hand gesture recognition performance using the KM [5], DCT [51], combined and DWT-FR features with RBF-NN classifier are tabulated in Table 3.3. The results in Table 3.3 show that the DWT-FR

Table 3.3: Comparative performances of hand gesture recognition using the KM, DCT, combined and DWT-FR feature sets with RBF-NN classifier for three distinct databases

Database	Feature set	Acc (%)	Sen (%)	Ppr (%)	Spe (%)
Database I	KM [5]	99.97	99.60	99.62	99.98
	DCT [51]	99.98	99.80	99.81	99.99
	Combined	99.97	99.60	99.62	99.98
	DWT-FR	**99.98**	**99.80**	**99.81**	**99.99**
Database II	KM [5]	99.34	92.08	92.23	99.66
	DCT [51]	99.74	96.92	96.96	99.87
	Combined	99.51	94.08	94.19	99.74
	DWT-FR	**99.77**	**97.25**	**97.42**	**99.88**
Database III	KM [5]	99.83	97.92	97.97	99.91
	DCT [51]	99.92	99.00	99.03	99.96
	Combined	99.85	98.25	98.28	99.92
	DWT-FR	**99.97**	**99.58**	**99.60**	**99.98**

features with RBF-NN classifier achieve an average accuracy, sensitivity, positive predictivity and specificity of 99.98%, 99.80%, 99.81% and 99.99% respectively for Database I, 99.77%, 97.25%, 97.42% and 99.88% respectively for Database II and 99.97%, 99.58%, 99.60% and 99.98% respectively for Database III. On the other hand, the KM features

with RBF-NN classifier show an average accuracy, sensitivity, positive predictivity and specificity of 99.97%, 99.60%, 99.62% and 99.98% respectively for Database I, 99.34%, 92.08%, 92.23% and 99.66% respectively for Database II and 99.83%, 97.92%, 97.97% and 99.91% respectively for Database III, the DCT features with RBF-NN classifier show an average accuracy, sensitivity, positive predictivity and specificity of 99.98%, 99.80%, 99.81% and 99.99% respectively for Database I, 99.74%, 96.92%, 96.96% and 99.87% respectively for Database II and 99.92%, 99.00%, 99.03% and 99.96% respectively for Database III, and the combined features with RBF-NN classifier show an average accuracy, sensitivity, positive predictivity and specificity of 99.97%, 99.60%, 99.62% and 99.98% respectively for Database I, 99.51%, 94.08%, 94.19% and 99.74% respectively for Database II and 99.85%, 98.25%, 98.28% and 99.92% respectively for Database III. The ROC graphs of hand gesture recognition using RBF-NN classifier with KM, DCT, combined and DWT-FR features for three distinct databases are shown in Figure 3.8. From Table 3.3 and Figure 3.8, it is observed that using RBF-NN classifier, the hand gesture recognition performance of the DWT-FR feature set is better compared to KM [5], DCT [51] and combined feature sets for all three gesture databases in all cases except only once case. For Database I, the gesture recognition performance of DWT-FR feature set is equal with DCT feature set and similarly, the gesture recognition performance of combined feature set is equal with KM feature set. However, in overall, the gesture recognition performance of DWT-FR feature set is better compared to KM, DCT and combined feature sets.

The average sensitivity of hand gesture recognition techniques using MLP-BP-NN and RBF-NN classifier with KM, DCT, combined and DWT-FR features for three hand gesture databases are shown in Figure 3.9. From Figure 3.9, it is observed that the hand gesture recognition performance using RBF-NN classifier is better compared to using MLP-BP-NN classifier for all three databases except the case of DWT-FR feature set for database I where the performances of both the classifiers are equal. From the above experimental study, it is concluded that the DWT-FR features provides better recognition performance compared to KM, DCT and combined features for hand gesture

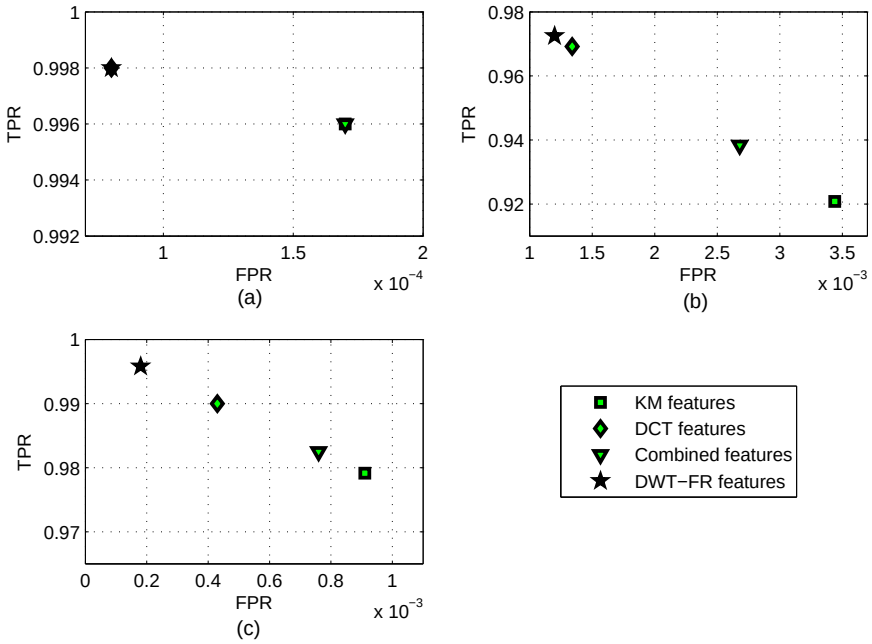


Figure 3.8: ROC graphs of hand gesture recognition using KM, DCT, combined and DWT-FR features with RBF-NN classifier for (a) Database I, (b) Database II and (c) Database III.

recognition. The DWT-FR feature set provides better recognition performance because it enjoys the following advantages: (i) the proposed feature extraction technique uses Haar wavelet which is very simple and follow the orthonormal property of wavelet basis [97]; (ii) the Haar wavelet effectively describes the soft parts of human body [111]; (iii) the DWT offers a simultaneous localization in time and spatial frequency domains; (iv) the DWT provides multi-resolution analysis of an image using higher level decomposition; and (v) the important DWT coefficient matrices are selected using an F-ratio based technique. From the above experimental study, it is also observed that the overall gesture recognition performance using RBF-NN classifier is better compared to MLP-BP-NN classifier. The RBF-NN classifier offers higher recognition performance because, in this network the localized radial basis functions are used to perform a nonlinear transform of input vector from input space to higher dimensional hidden space where vectors are more linearly separable and the RBF-NN is also free from local minima problem.

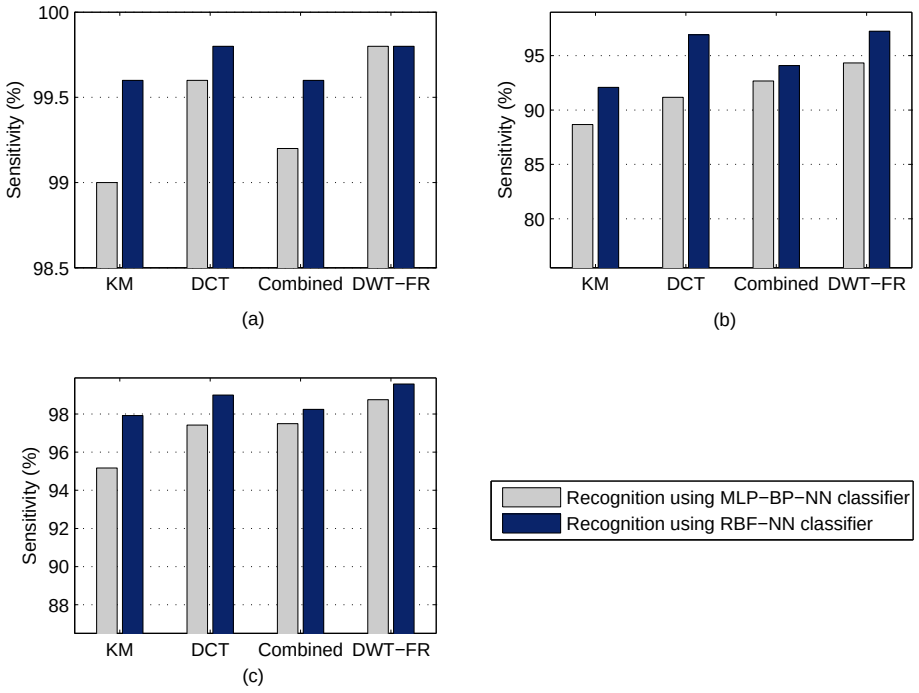


Figure 3.9: The average sensitivity of hand gesture recognition techniques using MLP-BP-NN and RBF-NN classifier with KM, DCT, combined and DWT-FR features for (a) Database I, (b) Database II and (c) Database III.

3.5 Conclusions

The overall conclusions in this chapter are as follows:

- This work proposes a feature extraction technique to extract the DWT and F-ratio based feature set for more efficient representation of hand gesture image.
- The proposed DWT and F-ratio based feature set represents the localized time and spatial frequency domain information of the gesture image. To obtain the DWT and F-ratio based feature set, the Haar wavelet is applied on the resized and enhanced grayscale image, and then the important DWT coefficient matrices are selected as features using the proposed F-ratio based technique. The RBF-NN is introduced in this chapter as classifier to classify static gesture images using

DWT and F-ratio based feature set for recognition.

- The experiments are separately conducted on three hand gesture databases using MLP-BP-NN and RBF-NN classifiers to evaluate the gesture recognition performance of the proposed DWT and F-ratio based feature set and to compare it with proposed combined features as shown in chapter 1 and other earlier reported techniques such as KM [5] and DCT [51] features.

- Experimental results demonstrate that in overall the proposed DWT and F-ratio based feature set offers better recognition performance compared to KM [5], DCT [51] and combined feature sets for all three databases. Experimental results also indicate that overall hand gesture recognition using RBF-NN classifier achieves higher recognition performance compared to MLP-BP-NN classifier.

CHAPTER 4

Hand Gesture Recognition using Modified RBF Neural Network Classifiers

4.1 Introduction

In chapter 3, standard RBF-NN classifier is used to recognize hand gesture images and provides higher recognition performance compared to MLP-BP-NN classifier. However, the selection of center in RBF-NN is one of the major issues for gesture recognition because the performance of the RBF-NN is dependent on selected centers [112]. In general, the centers of RBF-NN are selected randomly from training data set. Also, the weights of the RBF-NN are estimated using the pseudo inverse solution [80]. The performance of the hand gesture recognition decreases due to the inaccurate estimation of weight matrix. Our aim is to update the estimated weight matrix at the training phase such that the training error will be minimized. Here, the training error represents the error between score generated output and actual target of the network at the training phase. To implement this idea, the least-mean-square (LMS) algorithm is adopted to update the estimated weight matrix in training phase such that it minimizes the mean square error (MSE). This in turn helps to improve the gesture recognition performance.

Therefore, this chapter proposes a modified RBF-NN classifier i.e. k-mean and LMS-based RBF-NN classifier with Gaussian activation function (represented as KLMS-RBFNN-GAF classifier) for better recognition of hand gesture images. In KLMS-RBFNN-GAF classifier, the Gaussian function is used as activation function, the centers are automatically selected using k-mean algorithm and the estimated weight matrix is updated using LMS algorithm. However, one of the difficulties in RBF-NN is the use of Gaussian function because it is inefficient to estimate the similar patterns. Therefore, this work also proposes a sigmoidal function based modified RBF-NN classifier i.e. k-mean and LMS-based RBF-NN classifier with composite sigmoidal activation function (represented as KLMS-RBFNN-SAF classifier). In KLMS-RBFNN-SAF classifier, the activation function is formed using a set of composite sigmoidal functions, centers are automatically selected using k-mean algorithm and estimated weight matrix is further updated using LMS algorithm. The aim of the proposed KLMS-RBFNN-SAF classifier is to improve the recognition performance with the help of better estimation of the sim-

ilar training patterns. For both proposed classifiers, the selected centers and updated weight matrix are stored at the end of the training phase and used to recognize hand gesture images during testing. Note that, for both the proposed classifiers, LMS algorithm is used to update the estimated weight matrix due to following reasons: (i) If system is an adaptive linear combiner and the input and the desired response are available, the LMS algorithm is generally the best choice due to its simplicity and ease of computation; (ii) it does not demand off-line gradient estimations or repetitions of data; and (iii) it provides a stable and robust performance against different signal conditions. To evaluate the gesture recognition performance of the proposed KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers and to compare them with other reported classifiers such as MLP-BP-NN and RBF-NN, experiments are separately conducted on three hand gesture databases using KM, DCT, proposed combined and proposed DWT and F-ratio based (DWT-FR) feature sets. Experimental results indicate that proposed modified RBF-NN classifiers provide better recognition performance compared to other reported classifiers such as MLP-BP-NN and standard RBF-NN.

4.1.1 Organization of the Chapter

The rest of the chapter is organized as follows. In Section 4.2, the k-mean and LMS algorithms are described. Section 4.3 presents the framework of proposed RBF neural network classifiers in details. The experimental results and performances of the proposed techniques are discussed in Section 4.4. Finally, this chapter is concluded in Section 4.5.

4.2 Theoretical Background

4.2.1 K-mean Algorithm

The k-mean algorithm is a clustering algorithm which classifies the data points or patterns based on features into K number of groups, where K is a positive integer [113]. The grouping is done by minimizing the distances between data and the corresponding

cluster centroid. The k-mean algorithm has been applied in several applications such as information retrieval and image segmentation [114]. The operational flowchart of k-mean algorithm is shown in Figure 4.1. The details of k-mean algorithm is as follow.

1. *Initialization*: Initialize the K number of centroids for K clusters.

2. *Distance calculation*: Calculate the Euclidean distances of each data points to centroid of K clusters using (4.1).

$$\begin{aligned} D_{i,j} &= \sqrt{(x_{1,i}-\mu_{1,j})^2+(x_{2,i}-\mu_{2,j})^2+\cdots+(x_{M,i}-\mu_{M,j})^2} \\ &= \sqrt{\sum_{l=1}^{M}(x_{l,i}-\mu_{l,j})^2}, \quad i=1,2,\cdots,N, \\ & l=1,2,\cdots,M \quad and \quad j=1,2,\cdots,K. \end{aligned} \tag{4.1}$$

   where, N is the total number of data points, M is the dimension of data points, K is total number of clusters, $D_{i,j}$ is the Euclidean distances of i^{th} data points $\mathbf{x}_i$ to centroid of j^{th} clusters μ_j.

3. *Group assignment*: The group of data point $\mathbf{x}_i$ is in the cluster which has the minimum value of the Euclidean distance as presented in (4.2)

$$Group\, of\, \mathbf{x}_i \equiv \arg \min_{j=1,2,\cdots,K} D_{i,j}. \tag{4.2}$$

4. *Centroid calculation*: The centroid of each and every clusters are calculated using (4.3).

$$\mu_j = \frac{1}{Z}\sum_{p=1}^{Z}\mathbf{x}_p, \quad p=1,2,\cdots,Z \; and \; j=1,2,\cdots,K \tag{4.3}$$

   where, Z is the total number of data points in j^{th} cluster.

5. *Checking convergence*: In this step, the convergence of the algorithm is ensured by finding whether the centroid of each and every cluster converge to fixed value or not. If the centroid of each and every cluster converge to fixed value (i.e. the centroid of each and every cluster does not change anymore), the process stops,

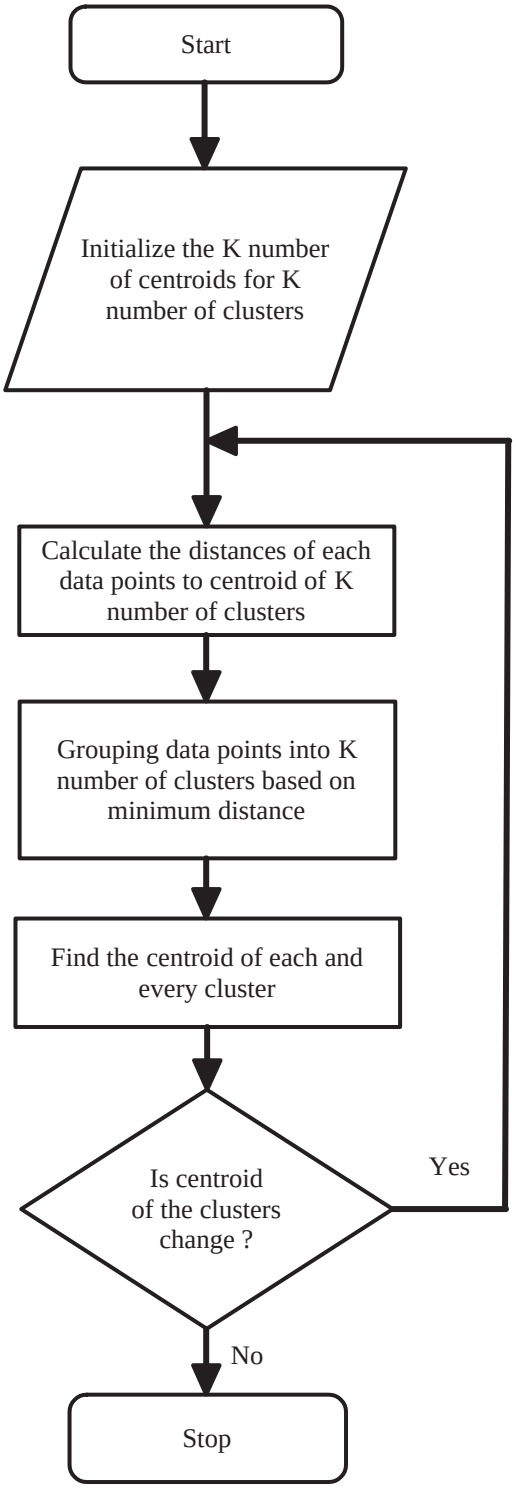


Figure 4.1: The operational flowchart of k-mean algorithm.

otherwise repeats from step 2 to step 5.

4.2.2 Least-Mean-Square (LMS) Algorithm

The least-mean-square (LMS) algorithm [80] is derived on the basis of instantaneous value of cost function, $\mathbf{w}$ which is defined in (4.4)

$$\xi(\mathbf{w}) = \frac{1}{2}e^2(k) \tag{4.4}$$

where, $e(k)$ represents the error signal measured at time k. The differentiation of $\xi(\mathbf{w})$ with respect to the weight vector $\mathbf{w}$ is expressed as

$$\frac{\partial \xi(\mathbf{w})}{\partial \mathbf{w}} = e(k)\frac{\partial e(k)}{\partial \mathbf{w}} \tag{4.5}$$

The LMS algorithm is worked similar to the least square filter and the error signal is defined as

$$e(k) = d(k) - \mathbf{x}^T(k)\mathbf{w}(k) \tag{4.6}$$

where, $d(k)$ and $\mathbf{x}(k)$ represents the desired signal and the input vector respectively measured at time k. The differentiation of $e(k)$ with respect to the weight vector $\mathbf{w}(k)$ is expressed as

$$\frac{\partial e(k)}{\partial \mathbf{w}(k)} = -\mathbf{x}(k). \tag{4.7}$$

Therefore, the equation (4.5) can be rewritten as

$$\frac{\partial \xi(\mathbf{w})}{\partial \mathbf{w}(k)} = -\mathbf{x}(k)e(k) \tag{4.8}$$

According to the steepest decent algorithm [80], the weight vector $\mathbf{w}(k+1)$ measured at time $(k+1)$ can be written as

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \eta\mathbf{g}(k) \tag{4.9}$$

where, η is a positive constant called the step-size or learning rate parameter and $\mathbf{g}(k)$ is the gradient vector. The gradient vector $\mathbf{g}(k)$ is defined as

$$\mathbf{g}(k) = \nabla\xi(\mathbf{w}) = \frac{\partial\xi(\mathbf{w})}{\partial\mathbf{w}(k)} = -\mathbf{x}(k)e(k). \tag{4.10}$$

Finally, the LMS algorithm is formulated using (4.9) and (4.10). The updating equation of the LMS algorithm can be defined as follows [80]:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \eta\mathbf{x}(k)e(k) \tag{4.11}$$

4.3 Proposed Framework

The final stage of a gesture recognition method is gesture classification. This chapter proposes two modified RBF-NN classifiers that are k-mean and LMS based RBF-NN with Gaussian activation functions (represented as KLMS-RBFNN-GAF classifier) and k-mean and LMS based RBF-NN with composite sigmoidal activation function (represented as KLMS-RBFNN-SAF classifier), for better recognition of hand gesture images. The details of proposed KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers are as follows.

4.3.1 K-mean and LMS based RBF-NN with Gaussian Activation Functions (KLMS-RBFNN-GAF)

This chapter first develops a k-mean and LMS based modified RBF-NN classifier where, Gaussian function is used as activation function to classify gesture images using extracted feature sets. The centers of the KLMS-RBFNN-GAF classifier are automatically selected using k-mean algorithm as presented in Algorithm 4.1 [9]. This center-selection algorithm selects an equal number of centers from training data of each hand gesture classes. This work uses k-mean algorithm to select centers of RBF-NN because (i) it is a unsupervised clustering algorithm which classifies the data points or patterns into K number of groups, where K is a positive integer [113]; (ii) it is simple, robust

and easier to understand [115] and (iii) the K number of centroids are selected as centers of RBF units for each hand gesture classes in a automatic manner. Initially, the

Algorithm 4.1: : Center-selection algorithm.

1: Let s be the number of gesture classes. Initially, set $i = 1$.
2: Let D_i be the training data of i^{th} class and k be the number of clusters in D_i. Initially, set first k number of samples from training data D_i as centroid of each cluster.
3: Compute the Euclidean distance between each sample of training data D_i and the centroid of each cluster. Assign each sample of training data D_i to the cluster according to nearest centroid.
4: Compute the centroid of each cluster.
5: Repeat steps (3)-(4), until centroid of the each cluster does not change anymore.
6: Store k number of centroids as centers of RBF neural network for i^{th} class and set $i = i + 1$.
7: Repeat steps (2)-(6), until $i <= s$.
8: Total $(k \times s)$ number of centers are selected for s classes.

weights $\mathbf{w}$ of the KLMS-RBFNN-GAF classifier are estimated using the pseudo inverse method [80]. The estimated weights are calculated by (4.12).

$$\mathbf{w} = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{t} \tag{4.12}$$

where, $\mathbf{G}^T$ is the transpose of $\mathbf{G}$; $(\mathbf{G}^T\mathbf{G})^{-1}$ is the inverse matrix of $(\mathbf{G}^T\mathbf{G})$; $\mathbf{t} = [t_1, t_2, \ldots, t_l]^T$ is target vector and $\mathbf{G} = [G_1(\mathbf{x}), G_2(\mathbf{x}), \ldots, G_i(\mathbf{x}) \ldots, G_m(\mathbf{x})]^T$ is projected input vector $\mathbf{x}$ in hidden space. Note that $G_i(\mathbf{x})$ is the i^{th} hidden node (or RBF unit) output. Here, the activation function $G_i(:)$ is chosen a Gaussian function and represented as

$$G_i(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_i\|}{\sigma^2}\right) \quad i = 1, 2, 3, ..., m \tag{4.13}$$

where $\mathbf{x}$ is an n-dimensional input vector, $\mathbf{c}_i$ is the center vector with the same dimension as $\mathbf{x}$, σ is the spread factor of i^{th} RBF unit and m is the number of hidden nodes (or number of centers) of the network. To improve the hand gesture recognition performance of the classifier, this work also proposes an LMS-based weights update technique which reduces the training error at the training phase. The training error represents the error

between score generated output and actual target of the network at the training phase. Our aim to update the estimated weight matrix based on LMS algorithm such that it minimizes the mean square error (MSE) in training phase. This in turn helps to improve the gesture recognition performance. The proposed weights update technique is given in Algorithm 4.2 [9]. The updated weight matrix and selected centers are stored at end of the training phase and used for testing purpose.

Algorithm 4.2: : The proposed weights update technique.

1: Initially, set number of iterations $n = 1$.
2: Set number of data points $j = 1$.
3: Compute estimated output vector $\mathbf{y}(j)$ for j^{th} data by multiplying estimated weights $\mathbf{w}(n)$ with $\mathbf{G}(\mathbf{x})$, where, $\mathbf{G}(\mathbf{x}) = [G_1(\mathbf{x}), G_2(\mathbf{x}), \ldots, G_m(\mathbf{x})]^T$ is projected input vector $\mathbf{x}$ at hidden space.
4: Compute the error vector which can be defined as $\mathbf{E}_1(j) = \mathbf{t}(j) - \mathbf{y}(j)$, where $\mathbf{t}$ and $\mathbf{y}$ are separately target vector and estimated output vector of the network. Assign $\mathbf{E}(n) = \mathbf{E}_1(j)$.
5: Update estimated weights using LMS algorithm [80] as follow

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta \mathbf{G}(\mathbf{x})\mathbf{E}(n)$$

where, n is iteration number, η is the learning rate parameter. Assign $\mathbf{w}(n) = \mathbf{w}(n+1)$.
6: Increase j by 1 and repeat steps (3)-(5) until j reaches N, where, N is the total number of data points.
7: Compute the mean square error which can be defined as $MSE(n) = \frac{1}{N}\sum_{j=1}^{N} \mathbf{E}_1^T(j)\mathbf{E}_1(j)$.
8: Increase n by 1 and repeat steps (2)-(7), until $MSE(n)$ is reduced to 0.0001 or n reach to 10000.

4.3.2 K-mean and LMS based RBF-NN with Composite Sigmoidal Activation Function (KLMS-RBFNN-SAF)

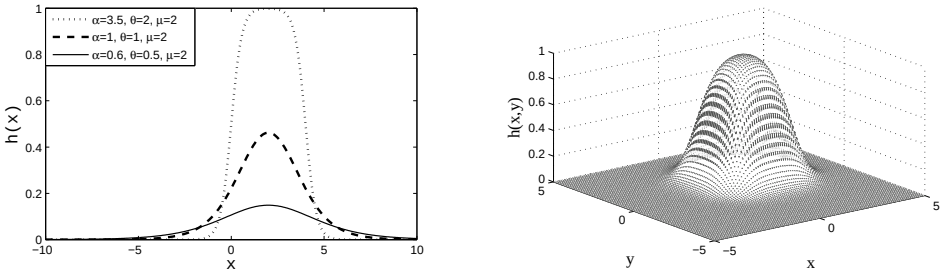
In KLMS-RBFNN-GAF classifier, Gaussian function is used as activation function. However, one of the difficulties in RBF-NN is the use of Gaussian function because it is inefficient to estimate the similar patterns. Therefore, to recognize hand gesture images, this work also proposes a sigmoidal function based modified RBF-NN classifier i.e. KLMS-RBFNN-SAF classifier. In KLMS-RBFNN-SAF classifier, the activation

function is formed using a set of composite sigmoidal functions, centers are automatically selected using k-mean algorithm and estimated weight matrix is further updated using LMS algorithm. The aim of the proposed KLMS-RBFNN-SAF classifier is to improve the recognition performance with the help of better estimation of the similar training patterns.

In one-dimensional case, a sigmoidal function is represented as $f(x) = (1 + e^{-\alpha(x-\mu)})^{-1}$, where $\alpha > 0$. The activation function for RBF network, by combining two sigmoidal functions [116] can be expressed as

$$h(x) = \frac{1}{1 + e^{-\alpha[(x-\mu)+\theta]}} - \frac{1}{1 + e^{-\alpha[(x-\mu)-\theta]}} \tag{4.14}$$

where $\alpha > 0$ and $\theta > 0$. Note that α, θ and μ represent shape parameter, shift parameter and center of the function $h(x)$ respectively. Three cases of function $h(x)$ with same center are shown in Figure 4.2(a). From Figure 4.2(a), it is observed that the shape of



(a) (b)

Figure 4.2: (a) Three different one-dimensional activation functions with same center. (b) A two-dimensional activation function with center vector = [1,1], shift parameter = 1.5, and shape parameter = 3.

$h(x)$ approximately rectangular for very high value of α and become Gaussian shape for low value of α. Therefore, $h(x)$ is a good candidate to efficiently estimate the training patterns. From Figure 4.2(a), it is also observed that $h(x)$ is maximum at μ and it has radial symmetry and local support property that meet the fundamental properties of the radial basis functions [116]. In higher dimensional case, the activation function

based on composite of sigmoidal functions can be defined as

$$h(\mathbf{x}) = \prod_{j=1}^{n} \left(\frac{1}{1+e^{-\alpha[(x_j-\mu_j)+\theta]}} - \frac{1}{1+e^{-\alpha[(x_j-\mu_j)-\theta]}} \right) \tag{4.15}$$

where $\mathbf{x} = [x_1, x_2, \cdots, x_n]^T$ is an input vector, $\mu = [\mu_1, \mu_2, \cdots, \mu_n]^T$ is the center vector, α is shape parameter and θ is bandwidth parameter of function $h(\mathbf{x})$. A two-dimensional case of (4.15) is shown in Figure 4.2(b). From (4.15), it is observed that the activation function for RBF network is formed using a set of composite sigmoidal functions and it can be considered as composite sigmoidal radial basis function (CSRBF). Using CSRBF as activation function, the RBF network for single output can be expressed in the following parametric from as presented in (4.16).

$$t(\mathbf{x}) = \sum_{i=1}^{m} w_i h_i(\mathbf{x}) \tag{4.16}$$

where

$$h_i(\mathbf{x}) = \prod_{j=1}^{n} \left(\frac{1}{1+e^{-\alpha[(x_j-\mu_{ij})+\theta]}} - \frac{1}{1+e^{-\alpha[(x_j-\mu_{ij})-\theta]}} \right) \tag{4.17}$$

is CSRBF output for i^{th} node, $\mathbf{x} \in R^n$ is input vector, α is shape parameter, μ_{ij} is the j^{th} element of the center vector in i^{th} node, θ is bandwidth parameter and m is the number of hidden nodes. Note that in RBF network, the number of input nodes, hidden nodes and output nodes are decided by dimension of input feature vector, number of centers and number of gesture classes respectively. The development of proposed RBF neural network is described as follow.

1. *Initialize the number of input and output nodes:* The number of input and output nodes of the RBF network are chosen equal as the dimension of input feature vector and number of gesture classes respectively.

2. *RBF centers selection:* The centers of the RBF network are selected by k-mean based center selection algorithm (as presented in Algorithm 4.1) depending on maximum recognition performance [2].

3. *Compute activation function output for each hidden node:* Here, CSRBF is used as activation function of RBF network and CSRBF output for i^{th} hidden node is computed using (4.17).

4. *Weight matrix estimation:* The weight matrix $\mathbf{w}$ of the RBF network is estimated the pseudo inverse solution [80]. The estimated weights are calculated by (4.18).

$$\mathbf{w} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{t} \tag{4.18}$$

   where $\mathbf{H}^T$ is the transpose of $\mathbf{H}$, $(\mathbf{H}^T\mathbf{H})^{-1}$ is the inverse matrix of $(\mathbf{H}^T\mathbf{H})$, $\mathbf{t} = [t_1, t_2, \ldots, t_l]^T$ is target vector and $\mathbf{H} = [h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_m(\mathbf{x})]^T$ is projected input vector $\mathbf{x}$ in hidden space.

5. *Update of estimated weights:* Our aim is to update the estimated weight matrix at the training phase such that the training error will be minimized. To implement this idea, LMS algorithm is adopted to update the estimated weight matrix in training phase such that it minimizes the mean square error (MSE). This in turn helps to improve the gesture recognition performance. The LMS-based weights update technique is given in Algorithm 4.2. The updated weight matrix and selected centers are stored at the end of training phase and used in testing phase.

4.4 Performance Evaluation

In this chapter, three hand alphabet databases of static gestures are used to evaluate the recognition performance of proposed modified RBF-NN classifiers that are KLMS-RBFNN-GAF classifier and KLMS-RBFNN-SAF classifier. As described in the earlier chapter, 50% data of each database is used to train the classifier and the remaining 50% is used to evaluate the recognition performance. The recognition performance is quantified using most common matrices [82] such as accuracy (Acc), sensitivity (Sen), positive predictivity (Ppr) and specificity (Spe). To analyze gesture recognition performance of KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers and to compare

their performances with existing classifiers such as MLP-BP-NN and standard RBF-NN, the experiments are separately conducted using KM, DCT, proposed combined and proposed DWT-FR features. The details of the experimental study are described as follows.

4.4.1 Experimental Results using Krawtchouk Moments (KM) Features

The experiments are separately conducted on three hand gesture databases using MLP-BP-NN, standard RBF-NN, KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers with KM features for performance analysis. The comparative performances of gesture recognition performance using MLP-BP-NN, standard RBF-NN, KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers with KM features are tabulated in Table 4.1. The comparative results show that the average accuracy, sensitivity, positive predictiv-

Table 4.1: Comparative performance of hand gesture recognition using MLP-BP-NN, standard RBF-NN, KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers with KM features for three distinct databases.

Database	Classifier	Acc (%)	Sen (%)	Ppr (%)	Spe (%)
Database I	MLP-BP-NN	99.92	99.00	99.05	99.96
	RBF-NN	99.97	99.60	99.62	99.98
	KLMS-RBFNN-GAF	**99.98**	**99.80**	**99.81**	**99.99**
	KLMS-RBFNN-SAF	**99.98**	**99.80**	**99.81**	**99.99**
Database II	MLP-BP-NN	99.06	88.67	89.09	99.51
	RBF-NN	99.34	92.08	92.23	99.66
	KLMS-RBFNN-GAF	**99.39**	**92.67**	**92.73**	**99.68**
	KLMS-RBFNN-SAF	**99.40**	**92.75**	**92.81**	**99.68**
Database III	MLP-BP-NN	99.60	95.17	95.61	99.79
	RBF-NN	99.83	97.92	97.97	99.91
	KLMS-RBFNN-GAF	**99.85**	**98.25**	**98.29**	**99.92**
	KLMS-RBFNN-SAF	**99.85**	**98.25**	**98.29**	**99.92**

ity and specificity of gesture recognition using KM features with MLP-BP-NN classifier are 99.92%, 99.00%, 99.05% and 99.96% respectively for Database I, 99.06%, 88.67%,

89.09% and 99.51% respectively for Database II and 99.60%, 95.17%, 95.61% and 99.79% respectively for Database III, with RBF-NN classifier are 99.97%, 99.60%, 99.62% and 99.98% respectively for Database I, 99.34%, 92.08%, 92.23% and 99.66% respectively for Database II and 99.83%, 97.92%, 97.97% and 99.91% respectively for Database III. On

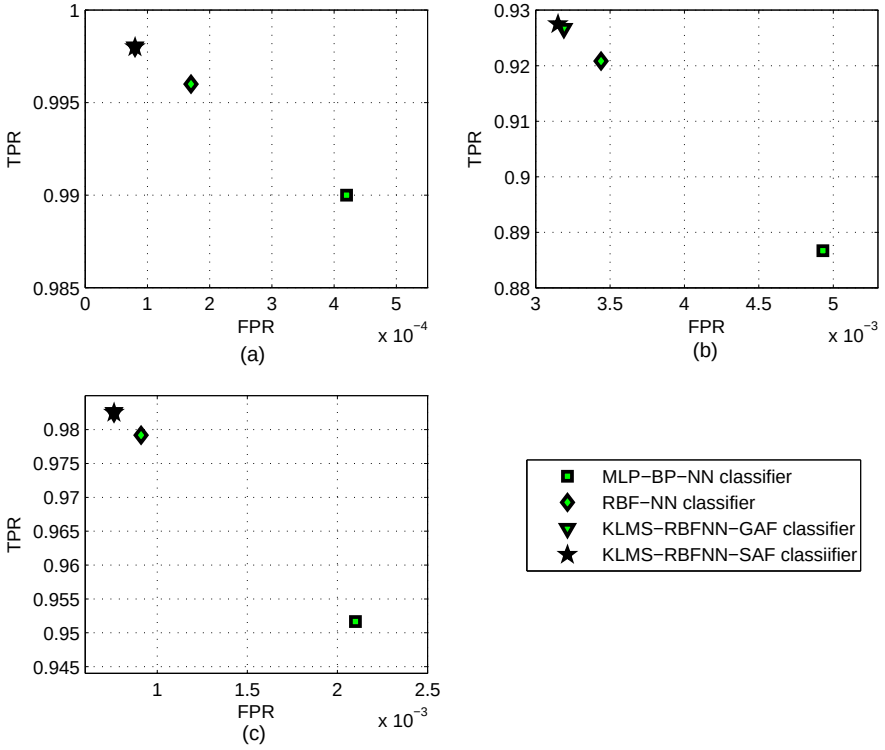


Figure 4.3: ROC graphs of gesture recognition using MLP-BP-NN, RBF-NN, KLMS-RBFNN-GAF, and KLMS-RBFNN-SAF classifiers with KM features for (a) Database I, (b) Database II and (c) Database III.

the other hand, the average accuracy, sensitivity, positive predictivity and specificity of gesture recognition using KM features with KLMS-RBFNN-GAF classifier are 99.98%, 99.80%, 99.81% and 99.99% respectively for Database I, 99.39%, 92.67%, 92.73% and 99.68% respectively for Database II and 99.85%, 98.25%, 98.29% and 99.92% respectively for Database III, and with KLMS-RBFNN-SAF classifier are 99.98%, 99.80%, 99.81% and 99.99% respectively for Database I, 99.40%, 92.75%, 92.81% and 99.68% respectively for Database II and 99.85%, 98.25%, 98.29% and 99.92% respectively for

Database III. The ROC graphs of gesture recognition using MLP-BP-NN, RBF-NN, KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers with KM features for three distinct databases are shown in Figure 4.3. From Table 4.1 and Figure 4.3, it is observed that using KM features, the modified RBF-NN classifiers that are KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers provide better recognition performance compared to MLP-BP-NN and RBF-NN classifiers for all three databases.

4.4.2 Experimental Results using Discrete Cosine Transforms (DCT) Features

The comparative gesture recognition performance of MLP-BP-NN, RBF-NN, KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers using DCT features for three hand gesture databases are shown in Table 4.2. From Table 4.2, it is observed that the

Table 4.2: Comparative performance of gesture recognition using MLP-BP-NN, standard RBF-NN, KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers with DCT features for three distinct databases

Database	Classifier	Acc (%)	Sen (%)	Ppr (%)	Spe (%)
Database I	MLP-BP-NN	99.97	99.60	99.62	99.98
	RBF-NN	99.98	99.80	99.81	99.99
	KLMS-RBFNN-GAF	**99.98**	**99.80**	**99.81**	**99.99**
	KLMS-RBFNN-SAF	**99.98**	**99.80**	**99.81**	**99.99**
Database II	MLP-BP-NN	99.35	92.17	92.27	99.66
	RBF-NN	99.74	96.92	96.96	99.87
	KLMS-RBFNN-GAF	**99.76**	**97.17**	**97.26**	**99.88**
	KLMS-RBFNN-SAF	**99.80**	**97.58**	**97.64**	**99.89**
Database III	MLP-BP-NN	99.83	98.00	98.06	99.91
	RBF-NN	99.92	99.00	99.03	99.96
	KLMS-RBFNN-GAF	**99.94**	**99.25**	**99.27**	**99.97**
	KLMS-RBFNN-SAF	**99.94**	**99.33**	**99.36**	**99.97**

average accuracy, sensitivity, positive predictivity and specificity of gesture recognition using DCT features with MLP-BP-NN classifier are 99.97%, 99.60%, 99.62% and 99.98% respectively for Database I, 99.35%, 92.17%, 92.27% and 99.66% respectively

for Database II and 99.83%, 98.00%, 98.06% and 99.91% respectively for Database III, and with RBF-NN classifier are 99.98%, 99.80%, 99.81% and 99.99% respectively for Database I, 99.74%, 96.92%, 96.96% and 99.87% respectively for Database II and 99.92%, 99.00%, 99.03% and 99.96% respectively for Database III. On the other hand,

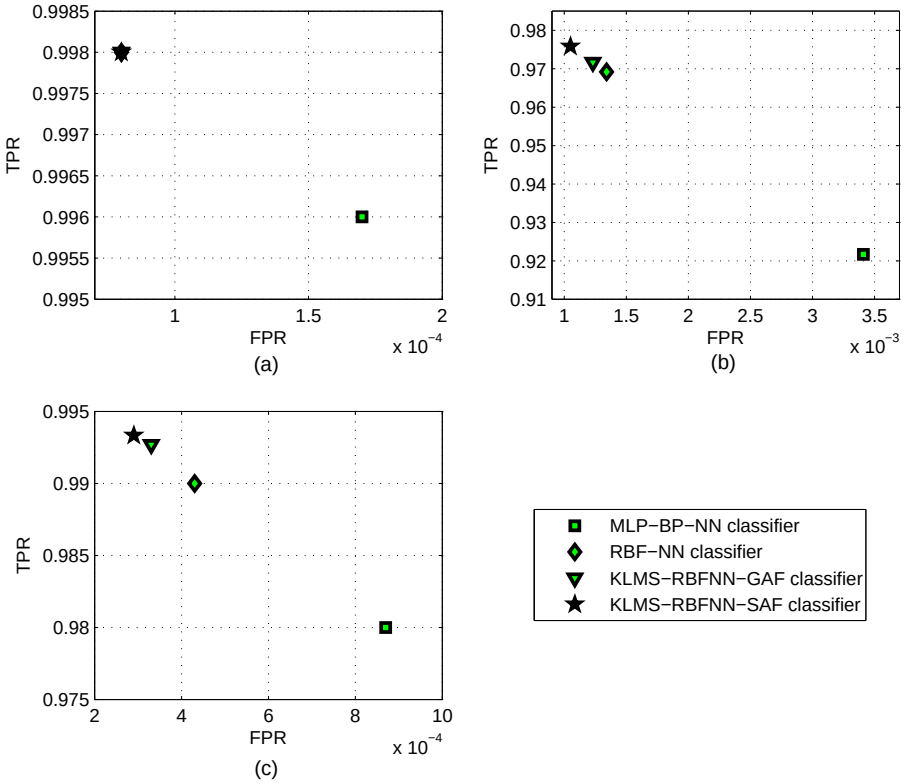


Figure 4.4: ROC graphs of gesture recognition using MLP-BP-NN, RBF-NN, KLMS-RBFNN-GAF, and KLMS-RBFNN-SAF classifiers with DCT features for (a) Database I, (b) Database II and (c) Database III.

the average accuracy, sensitivity, positive predictivity and specificity of gesture recognition using DCT features with KLMS-RBFNN-GAF classifier are 99.98%, 99.80%, 99.81% and 99.99% respectively for Database I, 99.76%, 97.17%, 97.26% and 99.88% respectively for Database II and 99.94%, 99.25%, 99.27% and 99.97% respectively for Database III, and with KLMS-RBFNN-SAF classifier are 99.98%, 99.80%, 99.81% and 99.99% respectively for Database I, 99.80%, 97.58%, 97.64% and 99.89% respectively for

Database II and 99.94%, 99.33%, 99.36% and 99.97% respectively for Database III. The ROC graphs of gesture recognition using MLP-BP-NN, RBF-NN, KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers with DCT features for three distinct databases are shown in Figure 4.4. Table 4.1 and Figure 4.3 indicate that using DCT features, the modified RBF-NN classifiers (KLMS-RBFNN-GAF and KLMS-RBFNN-SAF) offer higher recognition performance compared to MLP-BP-NN and RBF-NN for all databases except Database I where the performances modified RBF-NN classifiers are equal with RBF-NN classifier.

4.4.3 Experimental Results using Combined Features

The experimental results of gesture recognition performance for three hand gesture databases using MLP-BP-NN, RBF-NN, KLMS-RBFNN-GAF, and KLMS-RBFNN-SAF classifiers with proposed combined features are tabulated in Table 4.3. From Table 4.3, it is observed that the average accuracy, sensitivity, positive predictivity and specificity of gesture recognition using combined features with MLP-BP-NN classifier

Table 4.3: Comparative performance of gesture recognition using MLP-BP-NN, standard RBF-NN, KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers with combined features for three distinct databases

Database	Classifier	Acc (%)	Sen (%)	Ppr (%)	Spe (%)
Database I	MLP-BP-NN	99.95	99.40	99.43	99.98
	RBF-NN	99.97	99.60	99.62	99.98
	KLMS-RBFNN-GAF	**99.98**	**99.80**	**99.81**	**99.99**
	KLMS-RBFNN-SAF	**99.98**	**99.80**	**99.81**	**99.99**
Database II	MLP-BP-NN	99.33	92.00	92.14	99.65
	RBF-NN	99.51	94.08	94.19	99.74
	KLMS-RBFNN-GAF	**99.51**	**94.17**	**94.26**	**99.75**
	KLMS-RBFNN-SAF	**99.52**	**94.25**	**94.34**	**99.75**
Database III	MLP-BP-NN	99.79	97.50	97.55	99.89
	RBF-NN	99.85	98.25	98.28	99.92
	KLMS-RBFNN-GAF	**99.91**	**98.92**	**98.94**	**99.95**
	KLMS-RBFNN-SAF	**99.92**	**99.08**	**99.11**	**99.96**

are 99.95%, 99.40%, 99.43% and 99.98% respectively for Database I, 99.33%, 92.00%, 92.14% and 99.65% respectively for Database II and 99.79%, 97.50%, 97.55% and 99.89% respectively for Database III, and with RBF-NN classifier are 99.97%, 99.60%, 99.62% and 99.98% respectively for Database I, 99.51%, 94.08%, 94.19% and 99.74% respectively for Database II and 99.85%, 98.25%, 98.28% and 99.92% respectively for Database III. On the other hand, the average accuracy, sensitivity, positive predictivity and specificity

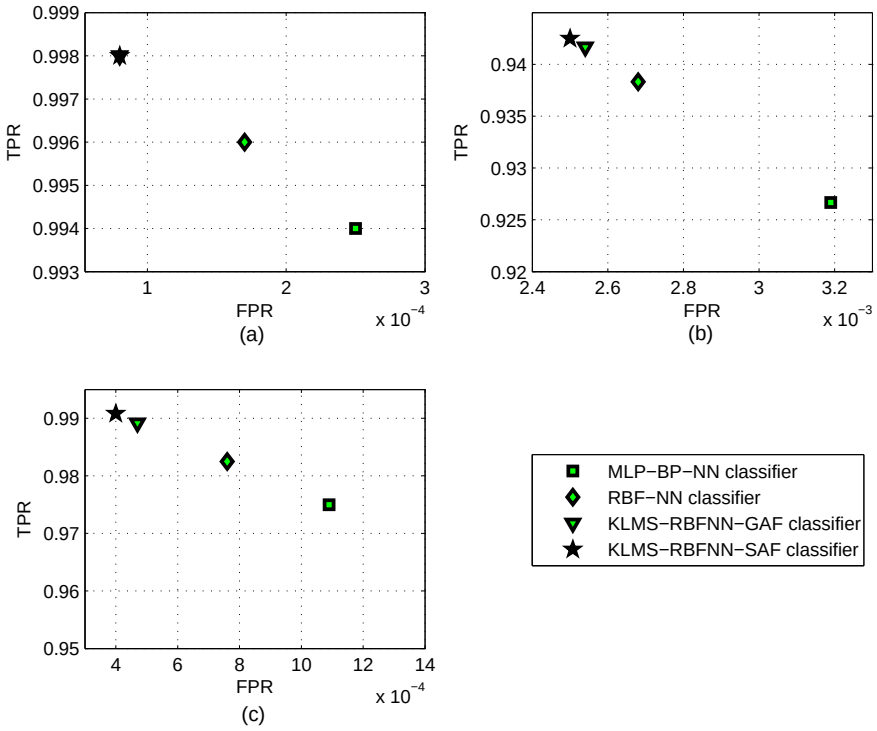


Figure 4.5: ROC graphs of gesture recognition using MLP-BP-NN, RBF-NN, KLMS-RBFNN-GAF, and KLMS-RBFNN-SAF classifiers with combined features for (a) Database I, (b) Database II and (c) Database III.

of gesture recognition using combined features with KLMS-RBFNN-GAF classifier are 99.98%, 99.80%, 99.81% and 99.99% respectively for Database I, 99.51%, 94.17%, 94.26% and 99.75% respectively for Database II and 99.91%, 98.92%, 98.94% and 99.95% respectively for Database III, and with KLMS-RBFNN-SAF classifier are 99.98%, 99.80%, 99.81% and 99.99% respectively for Database I, 99.52%, 94.25%, 94.34% and 99.75%

respectively for Database II and 99.92%, 99.08%, 99.11% and 99.96% respectively for Database III. For combined features, the ROC graphs of gesture recognition using MLP-BP-NN, RBF-NN, KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers for three distinct databases are shown in Figure 4.5. Table 4.3 and Figure 4.5 indicate that the recognition performances of modified RBF-NN classifiers (KLMS-RBFNN-GAF and KLMS-RBFNN-SAF) are better compared to MLP-BP-NN and RBF-NN classifiers for all three databases when combined features are used as input feature set.

4.4.4 Experimental Results using DWT and F-ratio based (DWT-FR) Features

The comparative gesture recognition performance of MLP-BP-NN, RBF-NN, KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers with proposed DWT-FR features for three hand gesture databases are tabulated in Table 4.4. From Table 4.4, it is noticed

Table 4.4: Comparative performance of gesture recognition using MLP-BP-NN, standard RBF-NN, KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers with DWT-FR features for three distinct databases

Database	Classifier	Acc (%)	Sen (%)	Ppr (%)	Spe (%)
Database I	MLP-BP-NN	99.98	99.80	99.81	99.99
	RBF-NN	99.98	99.80	99.81	99.99
	KLMS-RBFNN-GAF	**100.00**	**100.00**	**100.00**	**100.00**
	KLMS-RBFNN-SAF	**100.00**	**100.00**	**100.00**	**100.00**
Database II	MLP-BP-NN	99.53	94.33	94.49	99.75
	RBF-NN	99.77	97.25	97.42	99.88
	KLMS-RBFNN-GAF	**99.80**	**97.58**	**97.64**	**99.89**
	KLMS-RBFNN-SAF	**99.81**	**97.75**	**97.78**	**99.90**
Database III	MLP-BP-NN	99.90	98.75	98.78	99.95
	RBF-NN	99.97	99.58	99.60	99.98
	KLMS-RBFNN-GAF	**99.97**	**99.67**	**99.68**	**99.99**
	KLMS-RBFNN-SAF	**99.98**	**99.75**	**99.76**	**99.99**

that the average accuracy, sensitivity, positive predictivity and specificity of gesture recognition using DWT-FR features with MLP-BP-NN classifier are 99.98%, 99.80%,

99.81% and 99.99% respectively for Database I, 99.53%, 94.33%, 94.49% and 99.75% respectively for Database II and 99.90%, 98.75%, 98.78% and 99.95% respectively for Database III, and with RBF-NN classifier are 99.98%, 99.80%, 99.81% and 99.99% respectively for Database I, 99.77%, 97.25%, 97.42% and 99.88% respectively for Database II and 99.97, 99.58, 99.60 and 99.98% respectively for Database III. However, the av-

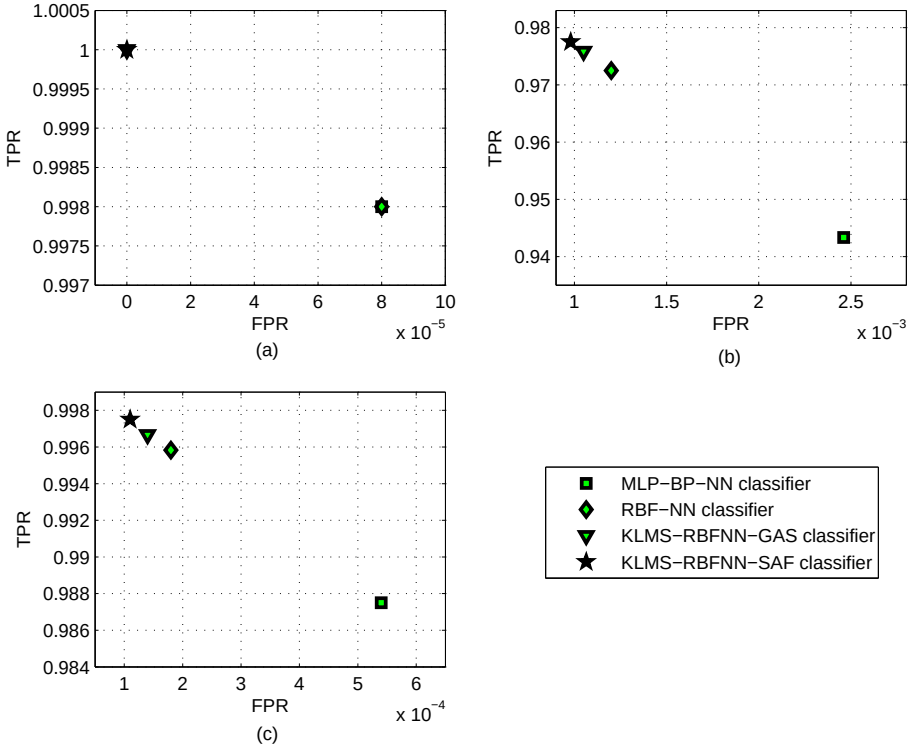


Figure 4.6: ROC graphs of gesture recognition using MLP-BP-NN, RBF-NN, KLMS-RBFNN-GAF, and KLMS-RBFNN-SAF classifiers with DWT-FR features for (a) Database I, (b) Database II and (c) Database III.

erage accuracy, sensitivity, positive predictivity and specificity of gesture recognition using DWT-FR features with KLMS-RBFNN-GAF classifier are 100.00%, 100.00%, 100.00% and 100.00% respectively for Database I, 99.80%, 97.58%, 97.64% and 99.89% respectively for Database II and 99.97%, 99.67%, 99.68% and 99.99% respectively for Database III, and with KLMS-RBFNN-SAF classifier are 100.00%, 100.00%, 100.00% and 100.00% respectively for Database I, 99.81%, 97.75%, 97.78% and 99.90% respec-

tively for Database II and 99.98%, 99.75%, 99.76% and 99.99% respectively for Database III. For proposed DWT-FR features, the ROC graphs of gesture recognition using MLP-BP-NN, RBF-NN, KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers for three distinct databases are shown in Figure 4.6. Table 4.4 and Figure 4.6 demonstrate that the gesture recognition performances of modified RBF-NN classifiers (KLMS-RBFNN-GAF and KLMS-RBFNN-SAF) are better compared to MLP-BP-NN and RBF-NN classifiers for all three databases when DWT-FR features are used as input feature set.

The average sensitivity of hand gesture recognition using MLP-BP-NN, RBF-NN, KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers with KM, DCT, proposed combined and proposed DWT-FR features for three hand gesture databases are shown in Figure 4.7. From Figure 4.7, it is observed that the performance of hand gesture recognition using proposed DWT-FR features is better compared to using KM, DCT, proposed combined features for all four different classifiers that are MLP-BP-NN, RBF-NN, KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers and all three different databases used except for two cases where the performances of DWT-FR features are equal with the performances of DCT features.

From the above experimental study it is demonstrated that in overall, gesture recognition performances of the proposed modified RBF-NN classifiers (KLMS-RBFNN-GAF and KLMS-RBFNN-SAF) are better compared to existing classifiers that are MLP-BP-NN and RBF-NN classifiers for all different feature sets like KM, DCT, proposed combined and proposed DWT-FR feature sets. The modified RBF-NN classifiers offer higher recognition performance because (i) they incorporate all the advantages of standard RBF-NN; (ii) they use a k-mean based center-selection algorithm which select the centers of RBF units in automatic manner; and (iii) they use an LMS based weights update technique to update the estimated weight matrix at training phase with the goal to minimize the MSE and this in turn helps to improve the gesture recognition performance. From the above experimental study, it is also demonstrated that the KLMS-RBFNN-SAF classifier offers better recognition performance compared to KLMS-RBFNN-GAF for all cases except for Database II using KM features and Database I. For

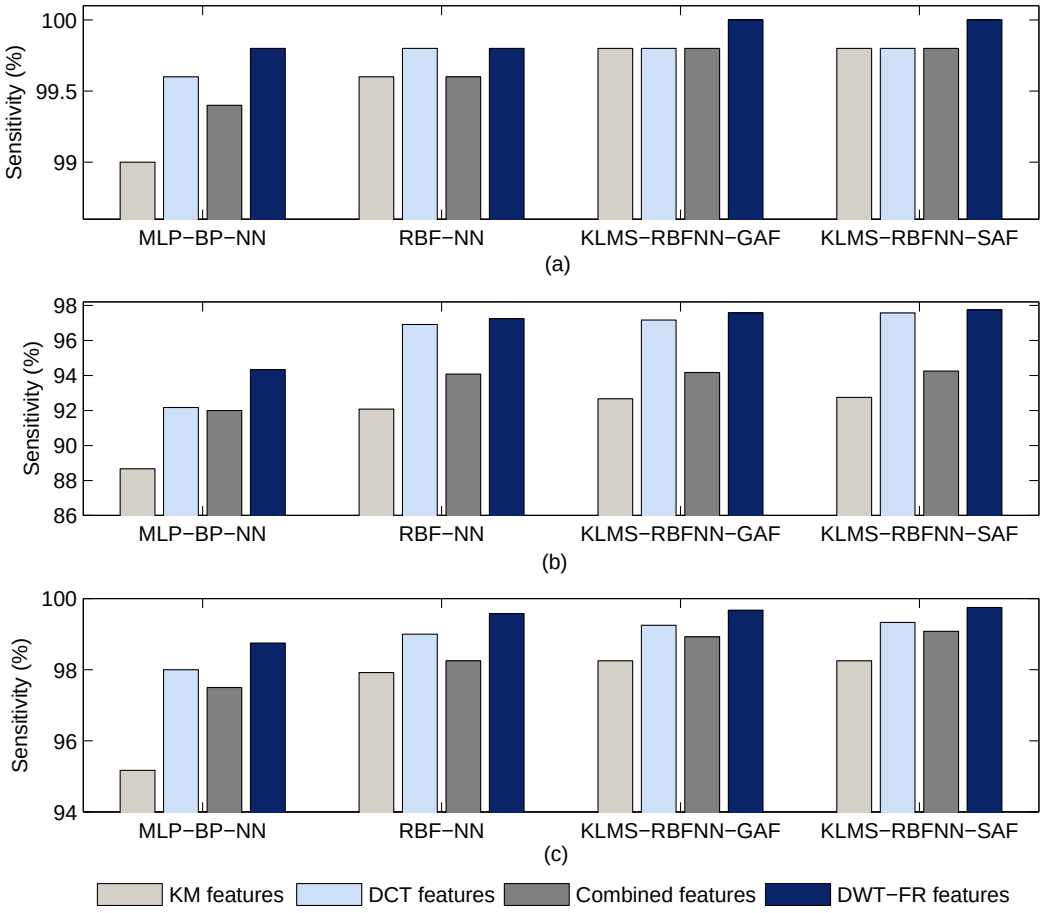


Figure 4.7: The average sensitivity of hand gesture recognition using MLP-BP-NN, RBF-NN, KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers with KM, DCT, proposed combined and proposed DWT-FR features for (a) Database I, (b) Database II and (c) Database III.

these cases, KLMS-RBFNN-SAF classifier provides an equal recognition performance with KLMS-RBFNN-GAF classifier. In most of the cases, the KLMS-RBFNN-SAF classifier provides better recognition performance over KLMS-RBFNN-GAF classifier because KLMS-RBFNN-SAF uses a set of composite sigmoidal functions as activation function which estimate the similar training patterns better compared to Gaussian activation functions.

4.5 Conclusions

The salient points covered in this chapter are:

- This chapter proposes two modified RBF-NN classifiers that are KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers for better recognition of hand gesture images.

- In KLMS-RBFNN-GAF classifier, the Gaussian function is used as activation function, the centers are automatically selected through k-means based center-selection algorithm and the initial trained weight matrix is further updated using LMS algorithm whereas in KLMS-RBFNN-SAF, activation function is formed using a set of composite sigmoidal functions, centers are automatically selected using k-mean algorithm and initial trained weight matrix is further updated using LMS algorithm. For both cases, selected centers and updated weight matrix are stored at the end of training phase and these are used during testing phase to recognize hand gesture images.

- The experiments are separately conducted on three hand gesture databases using KM, combined, DCT and DWT-FR features to evaluate gesture recognition performance of the proposed KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers and to compare them with other reported classifiers such as MLP-BP-NN and RBF-NN.

- Experimental results demonstrate that the proposed modified RBF-NN classifiers provides higher recognition performance compared to other reported classifiers such as MLP-BP-NN and RBF-NN. Experimental results also indicate that the KLMS-RBFNN-SAF classifier offers better or equal recognition performance compared to KLMS-RBFNN-GAF classifier.

- From experimental results, it is also observed that the performance of hand gesture recognition using DWT-FR features is better compared to using KM, DCT

and proposed combined features for all different classifiers including MLP-BP-NN, RBF-NN, KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers.

CHAPTER 5

Feature Vector Optimization using Genetic Algorithm for Hand Gesture Recognition

5.1 Introduction

In general, the feature subset selection (FSS) has been used in the area of pattern recognition where large datasets are involved [50]. A major problem associated with pattern recognition is the "curse of dimensionality" in which too many features are used to represent the patterns [117]. This problem led to a large number of classifier parameters (e.g., weights of MLP-BP-NN) [118] which in turn increases computation for pattern classification. Therefore, the reduction of feature dimensionality has become an important task in the areas of pattern recognition [119], classification [120], data mining [121], bioinformatics [122] etc. The reduction of feature dimensionality contributes several advantages [50] for a recognition system employed for a specific application: (i) a reduction in the cost of acquisition of the data, (ii) improvement of the understandability of the final classification model, (iii) a faster induction of the final classification mode, (iv) an improvement in recognition performance. Feature Selection (FS) can be defined as a process to select a subset of features from original feature set. This can be carried out through eliminating the redundant, uninformative, and noisy features [50]. By reducing feature vector dimensionality, feature selection process not only reduce the cost of recognition method but also in some cases it can provide a better classification performance due to finite sample size effects [123].

In this chapter, a genetic algorithm (GA) based feature vector optimization technique is adopted to optimize the original feature set for hand gesture recognition [9]. The objective of the feature vector optimization is to reduce the dimension of feature vector by selecting an optimal subset of features that contains the most informative and discriminative information of patterns. The length of extracted feature set is reduced by eliminating the unnecessary, redundant and irrelevant features. This work uses GA for feature vector optimization because it has following advantages: (i) it is a randomized search and optimization technique guided by the principles of evolution and natural genetics; (iii) For an optimization problem, GA performs search in complex, large and multimodal landscapes, and provide near-optimal solutions [124]; (iv)

it is considered to be a robust method because no restrictions on the solution space are made during the process; (v) it exploits historical information structures from the previous solution to future solution which in turn to increase the performance of future solution structures [125]; (vi) it is an inherently parallel optimization method where the calculations of the fitness function on all search points of a population are completely independent and can be carried out in several operations. Therefore, rather than being trapped in a suboptimal local maximum or minimum, it can move to near global optimum solution [126, 127]. In this chapter, three hand alphabets databases of static gesture are used for recognition. As described in chapter 2, a common training and testing dataset is used for performance evaluation. The extracted sets of KM, DCT, proposed combined and proposed DWT-FR features are separately optimized using GA with proposed KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers. The KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers are discussed in detail in chapter 4. The final optimized feature subsets of KM, DCT, proposed combined and proposed DWT-FR features are applied to the input of the proposed classifiers. The experimental results indicate that the proposed technique significantly reduces the dimension of feature vectors which in turn helps to reduce feature space complexity and for most of the cases, it also provides a better performance for hand gesture recognition.

5.1.1 Organization of the Chapter

The rest of the chapter is organized as follows. The theoretical background of the genetic algorithm (GA) is discussed in Section 5.2. The framework of optimum feature selection technique is thoroughly described in Section 5.3. The experimental results and performance comparisons are presented in Section 5.4. Finally, the chapter is concluded with some final remarks in Section 5.5.

5.2 Theoretical Background

5.2.1 Genetic Algorithm (GA)

The GA is a randomized search and optimization algorithm inspired by biological evolution process and natural genetics [124]. In GA, genetic operations such as selection, crossover and mutation are used for the strategy of problem-solving. The basic theory of GAs are based on Darwin's theory of evolution "Survival of the fittest" and follows the natural evolution law i.e. the strong survivors have better opportunity to transfer their genes to future generations through reproduction [128,129]. Species those convey better gene combination are dominant in the population. In evolution process, sometimes random changes may occur in genes. New species evolve from the old ones if the gene changes deliver more favourableÂăposition for the survival [129]. On the other hand, the species which does not success to change genes, are eliminated through the natural selection.

Primitively the GA was investigated by J. Holland in 1975 [130]. Later on, the GA became popular after publishing the book by D. Goldberg in 1989 [131]. In this book, the author has described the theory, operation, and application of genetic algorithms. The GAs have been used for application in several fields such as pattern recognition [124], bioinformatics [132], control engineering [133,134], radar pulse compression [129], traveling salesman problem [135,136] etc. For a given problem, GAs maintain a set of candidate solutions known as population and iterate on the complete population. Each iteration is named as a generation. In the GA, a single solution point in the problem space is represented by a fixed length string of symbols. This string is referred to as a chromosome, the symbols are referred to as genes and the values of genes are called alleles [125]. The details of various steps which are involved in GA are as follows.

5.2.1.1 Initialization of Population

The population size is a user-specified parameter and it indicate the number of chromosomes in one generation. The performance of GAs is affected by the size of the

population. If population size is too small, the possibility of performing crossover is less and only a small number of solution points are explored in search space. Therefore, it provides a substandard solution for a given problem. On the other hand, too many chromosomes increase the computational time and slow down the GA process. The representation of a chromosome should be in such a way that it contains information about the solution. In general, the chromosomes are represented in real numbers such as -0.6, 0.3, 1.2 etc or in a string of binary bits such as '101010011011', '110110100101', '100111001001' etc. which are encoded using an coding process [129]. In this work, chromosomes are represented in a string of binary bits. All the chromosomes in a population are randomly initialized with a fixed length of the binary bit string.

5.2.1.2 Evaluation of Fitness Function

The fitness function is an objective function or a mathematical model to be optimized [136]. The fitness values of the candidate solutions or chromosomes of the initialized population or created offspring population are evaluated using the fitness function. These values are used to distinguish good chromosomes from bad chromosomes and also help to implement natural selection.

5.2.1.3 Selection

Selection is a mechanism that can be used to select chromosomes from the population and enter into a mating pool. The chromosomes of the mating pool are involved to produce offspring for the next generation. Since the genes of the chromosome from current generation are transformed to the next generation inheritable in manner, therefore the chromosomes of the mating pool should be good according to fitness values. In GA, the selection procedure is used to impose the survival-of-the-fittest mechanism on the candidate solutions by selecting more number of good chromosomes from the population for the mating pool [137]. Therefore, the selection procedure helps GA to improve the fitness of candidates in the population over succeeding generations. There are several selection techniques in the literature such as roulette-wheel selection, steady state

selection, ranking selection and tournament selection [138] etc. In this thesis roulette-wheel selection is used to select a chromosome from the population. In roulette-wheel selection, each individual in the population is assigned a slot or portion of a roulette wheel in proportion to its fitness, where the sum of all fitness values of the individuals in the population indicates the entire circumference of the roulette wheel. Therefore, the least fit individual occupies smaller segment, whereas the fittest individual occupies the largest segment within the roulette wheel. In roulette-wheel selection, each chromosome in the population has a chance to select and enter into a mating pool as a parent chromosome with a probability that is proportional to its occupied segment size within the roulette wheel. Therefore, the fittest chromosomes will be chosen more frequently than least fit ones and satisfy the demands of survival-of-the-fittest. At the same time, the roulette-wheel selection preserves the diversity in the population because it gives a selection opportunity to all of the individuals in the population instead of discarding none of them [136]. The algorithm of the roulette-wheel selection procedure [138] is given as follows.

1. Let, $x_1, x_2, x_3, \cdots, x_N$ be the chromosomes in the population, where N is the population size. Calculate the fitness values of each and every chromosome in the population.

2. Compute the selection probability (segment size) of each individual of the population using (5.1).

$$P_i = \frac{F_i}{\sum_{j=1}^{N} F_j}, \qquad i = 1, 2, 3, \cdots, N \tag{5.1}$$

   where, P_i and F_i represent the selection probability and fitness value of i^{th} chromosome respectively.

3. Compute the cumulative probability for each individual using (5.2).

$$Q_i = \sum_{j=1}^{N} P_j, , \qquad i = 1, 2, 3, \cdots, N, \tag{5.2}$$

where, Q_i is the cumulative probability of i^{th} chromosome.

4. Generate a uniform random number, r, where $r \in (0, 1]$.

5. If $r < Q_1$, then select the first chromosome x_i for mating pool, otherwise select the i^{th} chromosome, x_i when $Q_{i-1} < r \leq Q_i$.

6. Repeat N times step 4 to step 5 to create the N number of chromosomes in the mating pool.

5.2.1.4 Crossover

In this operation, two parental chromosomes, selected from the existing population are combined together to create new chromosomes. These newly created chromosomes are called offspring. It is expected that the chromosomes in offspring population inherit the good genes from parental chromosomes. The crossover operation is accomplished with a certain probability, p_c, called as crossover probability which indicates how frequently crossover will be performed [138]. The offspring chromosomes are formed from the parts of the parent population. If the value of crossover probability is 1, all chromosomes in offspring population will be formed through crossover operation. On the other hand, if the value of crossover probability is 0, unlike in crossover operation, all chromosomes in offspring population will be the exact copies of the chromosomes from old population. The crossover operation is carried out expecting the offspring chromosomes will be better from parental chromosomes. However, it is good to leave some part of the parent population to survive for next generation. There are several crossover mechanisms such as single point crossover, two point crossover and uniform crossover which are involved in recombination process [137]. The mechanisms of a single point crossover, a two point crossover and an uniform crossover operation are shown in Figure 5.1. In the case of single point crossover as shown in Figure 5.1 (a), a crossover point is selected randomly over the string length and all the genes after this point are swapped between the pair of parental chromosomes to create two offspring chromosomes. Similarly, for two point crossover, illustrated in Figure 5.1 (b), two crossover points are

selected randomly over the string length and the genes in between the two crossover points are swapped between the pair of parental chromosomes to create two offspring chromosomes. In case of uniform crossover, illustrated in Figure 5.1 (b), every gene is swapped between the pair of parental chromosomes with a certain probability, p_s, known as the swapping probability to create two offspring chromosomes. In general, the value of swapping probability is chosen as 0.5 [137].

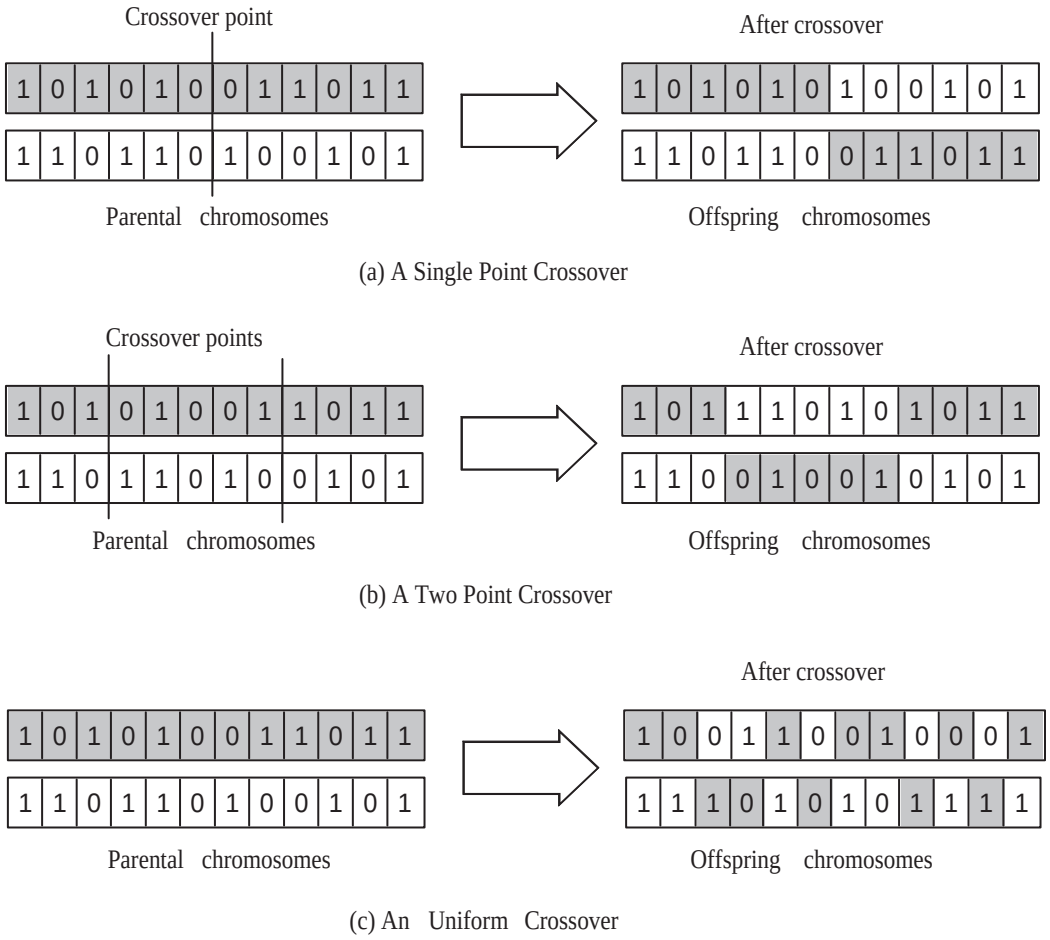


Figure 5.1: The mechanisms of (a) a single point crossover, (b) a two point crossover and (c) an uniform crossover operation.

5.2.1.5 Mutation

This operation is performed at gene level to change the features of chromosomes. Mutation is designed to add diversity to the population and ensure the possibility to search the entire search space. Unlike crossover, mutation is the secondary operator in GAs and carried out with a low probability, p_m, known as mutation probability [138]. If there is no mutation, the offspring is just taken after crossover (or copy) without any change whereas part of the chromosome is changed when a mutation is performed. If the value of mutation probability is 1, then the whole chromosome will be changed and if it is 0, nothing will be changed. Mutation reintroduces the genetic diversity of chromosomes in the population and helps to avoid being stuck in a local minima solution [129]. The most common mutations strategy is the bit-flip mutation where randomly selected bit in a binary string is switched (a 1 to 0 or 0 to 1) based on mutation probability. One example of mutation operation is shown in Figure 5.2.

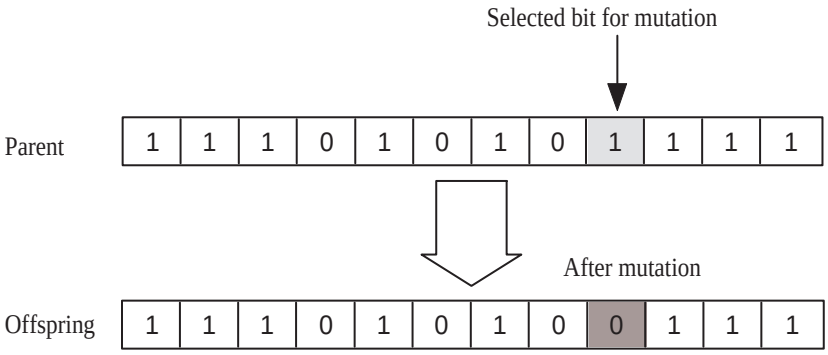


Figure 5.2: Mutation operation.

5.2.1.6 Replacement

When the offspring chromosomes are created using selection, recombination, and mutation operation, they are used to replace the parental chromosomes. There are many replacement techniques such as elitist replacement, delete-all replacement and steady-state replacement which are used in GAs [138]. In elitist replacement, the worst parental chromosomes in current population are replaced by best newly created offspring

chromosomes [139]. In delete-all replacement, all the parental chromosomes in current population are replaced by offspring chromosomes that have just been created. In steady-state replacement, n number of parental chromosomes are replaced by n number of new offspring chromosomes where the value of n should be smaller than the population size. This method involves two considerations: which member and how many members are to be replaced from the current population [138]. In this thesis, elitist replacement technique [139] is used to replace the parental chromosomes by offspring chromosomes because it ensures that the chromosomes in the new population are more fit compared to the chromosomes in old population.

5.2.1.7 Termination

In general, the process of genetic operations including selection, crossover, mutation and replacement is terminated when generation reaches it's predefined maximum generation [138]. The process stops and finds the best solution, if termination criteria is satisfied; otherwise, it proceeds for next generation. The flowchart of GA operation is shown in Figure 5.3.

5.3 Proposed Framework: Optimum Feature Subset Selection using Genetic Algorithm (GA)

The dimension of extracted feature set is reduced using GA which removes unwanted features and simultaneously selects the optimum feature subset from original feature set. The optimized selected feature subset is finally applied to input of the proposed KLMS-RBFNN-SAF classifier to recognize the hand gesture images. In GA based feature subset selection technique, each chromosome in the population denotes one possible solution for feature subset selection. The dimension of search space which represents the number of genes of the chromosome is equal to the length of the extracted proposed DWT-FR feature set. In this thesis, each chromosome in the population is represented by a string of binary bits where "0" or "1" denotes the value of genes. The direct binary coding

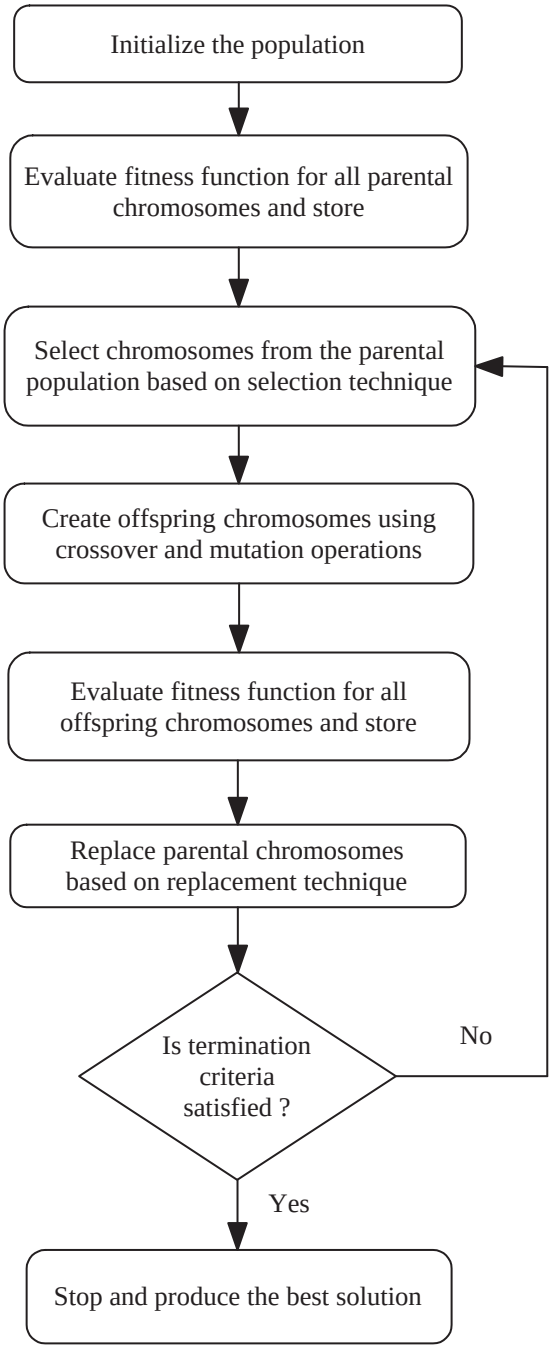


Figure 5.3: The flowchart of GA.

scheme is used here to select feature subset from original feature set. In direct binary coding scheme, as shown in Figure 5.4, if value of gene or bit is 1 then corresponding feature will be selected and if it is 0 then not-selected [9]. In each generation, GA creates new offspring from parental chromosomes for the new possible solution of feature subset selection. In this work, the mean square error in training phase and number of reduced

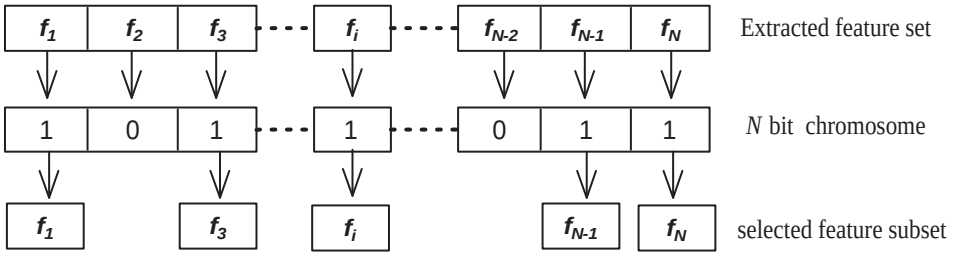


Figure 5.4: Feature subset selection using chromosome bit pattern.

features are used to design a fitness function which is defined [9] as

$$F = 0.8 \times (MSE_{train})^{-1} + 0.2 \times (N - N_s) \tag{5.3}$$

where, MSE_{train}, N, N_s and $(N - N_s)$ represent the mean square error in the training phase, the total number of features in extracted feature set, the number of selected features and the number of reduced features respectively. The values of weights for MSE_{train} and the number of reduced features (i.e. $N - N_s$) in fitness function are chosen empirically as 0.8 and 0.2 respectively. From the fitness function as defined in (5.3), it is seen that for a chromosome (possible solution for feature subset selection), the high fitness value is obtained when MSE_{train} is low and the number of reduced features is high. Note that the lower MSE_{train} helps to improve the classification performance whereas a higher number of feature reduction shows lower dimensional complexity in feature space for final classification. The chromosome with high fitness value is more appropriate to survive for next generation and provide the better solution for feature subset selection. The flowchart of optimum feature subset selection technique using GA is shown in Figure 5.5. The detailed procedure of optimum feature subset selection using

GA is as follows.

1. **Initial population:** Initially, randomly generate N_p number of chromosomes, where each chromosome is a binary bit pattern of N bits and N_p represents the population size. Note that the value of N is equal to the length of the proposed DWT-FR feature set.

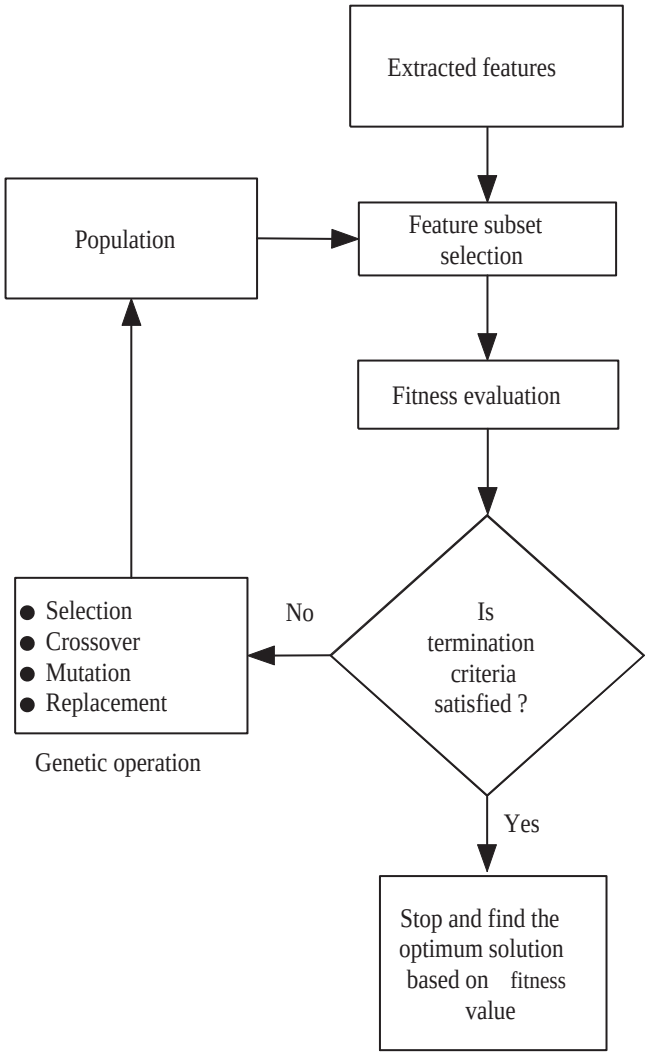


Figure 5.5: The flowchart for GA-based feature subset selection technique.

2. **Feature subset selection:** For each chromosome bit pattern, a feature subset is selected using direct binary coding scheme as shown in Figure 5.4.

3. **Fitness evaluation:** For each chromosome, the selected features are used to train the proposed KLMS-RBFNN-SAF classifier and corresponding mean square error (MSE) is calculated. The fitness value of the chromosome is calculated using the fitness function as defined in (5.3).

4. **Termination criteria:** When the termination criteria (reaching of maximum generation) is satisfied, the process stops and finds the optimum solution which gives the maximum fitness value; otherwise, it proceeds for the next generation.

5. **Genetic operation:** In this step, the system searches for better solutions by genetic operations, including selection, crossover, mutation, and replacement.

   (a) **Selection operation:** The roulette-wheel selection scheme is used here to select chromosomes for mating pool population. The chromosomes of mating pool population are involved to produce offspring for the next generation through crossover and mutation operations.

   (b) **Crossover operation:** In this work, the crossover operation is carried out by empirically choosing the crossover probability as 0.7. This indicates that around 70% of mating pool chromosomes are involved in crossover operation. A uniform crossover operation with 0.5 swapping probability is used here to generate offspring chromosomes.

   (c) **Mutation operation:** In this work, the bit-flip mutation strategy is used to switch randomly selected bit in a binary string with empirically choosing 0.1 mutation probability. This indicates that around 10% bits of the chromosome are flip from 1 to 0 or 0 to 1. After crossover and mutation operation, an offspring population is obtained from mating pool population.

   (d) **Replacement operation:** Once the offspring population is produced, the fitness value of all offspring chromosomes are calculated based on step 2 and step 3. In this work, using elitist replacement technique, a parental population is produced by selecting best N_p number of chromosomes from mating pool and offspring population for next generation.

At the end of processes, when termination criteria is satisfied, the bit pattern of the best chromosome is used to select the optimum feature subset from original extracted feature set. In the training phase, the training dataset with optimized feature set is further used to train the classifier. On the other side, during testing, the selected optimized feature set based on the bit pattern of the best chromosome is used to recognize the hand gesture image.

5.4 Performance Evaluation

Three databases of static hand gesture are considered in this thesis for performance evaluation of proposed feture vector optimization technique. As described in chapter 2, the 50% of randomly selected data of each database are used as train dataset and the remaining 50% data are used as test patterns for performance evaluation. The recognition performance is evaluated using accuracy (Acc), sensitivity (Sen), positive predictivity (Ppr) and specificity (Spe) parameters [82]. In chapter 4, the gesture recognition performance of the proposed KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers are compared with existing classifiers such as MLP-BP-NN and standard RBF-NN using KM, DCT, proposed combined and proposed DWT-FR features. In this chapter, the extracted feature sets of each database are optimized individually using GA with proposed KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers. The detail setting of parameters for GA which are empirically chosen as follows: population size (N_p):20, crossover probability (c_p):0.7 and mutation probability (m_p):0.1. The uniform crossover, roulette-wheel selection and elitism replacement techniques are used here. The length of the chromosome (N) is taken as same as the length of extracted feature set and the maximum number of generation is assigned as 50. To evaluate the performance of GA based feature optimization technique, the experiments are separately conducted using KM, DCT, proposed combined and proposed DWT-FR features with proposed KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers.

5.4.1 Experimental Results using KLMS-RBFNN-GAF Classifier

The performance of the GA based feature vector optimization technique with KLMS-RBFNN-GAF classifier for KM, DCT, proposed combined and proposed DWT-FR feature sets are shown in Table 5.1. From Table 5.1, it is observed that the lengths of

Table 5.1: Performance of GA based feature vector optimization technique with KLMS-RBFNN-GAF classifier for KM, DCT, proposed combined and proposed DWT-FR feature sets.

Database	Feature set	Length of feature sets		% of reduction
		Original	Optimized	
Database I	KM	64	**18**	**71.88**
	DCT	100	**30**	**70.00**
	Combined	163	**104**	**36.20**
	DWT-FR	192	**50**	**73.96**
Database II	KM	64	**39**	**39.06**
	DCT	100	**52**	**48.00**
	Combined	313	**148**	**52.72**
	DWT-FR	192	**106**	**44.79**
Database III	KM	64	**39**	**39.06**
	DCT	100	**51**	**49.00**
	Combined	421	**228**	**45.84**
	DWT-FR	192	**90**	**53.13**

optimized KM, DCT, proposed combined and proposed DWT-FR feature sets are 18, 30, 104 and 50 respectively for Database I, 39, 52, 148 and 106 respectively for Database II, and 39, 51, 228 and 90 respectively for Database III whereas the lengths of original KM, DCT, proposed combined and proposed DWT-FR feature sets are 64, 100, 163 and 192 respectively for Database I, 64, 100, 313 and 192 respectively for Database II and 64, 100, 421 and 192 respectively for Database III. Therefore, for KM, DCT, proposed combined and proposed DWT-FR feature sets, the GA based feature vector optimization technique with KLMS-RBFNN-GAF classifier reduces number of features as 71.88%, 70.00%, 36.20% and 73.96% respectively for Database I, 39.06%, 48.00%, 52.72% and

44.79% respectively for Database II and 39.06%, 49.00%, 45.84% and 53.13% respectively for Database III.

The performance of hand gesture recognition using KLMS-RBFNN-GAF classifier with original and GA based optimized KM, DCT, proposed combined and proposed DWT-FR feature sets for three different databases are shown in Table 5.2. The results in Table 5.2 indicate that the average sensitivities of hand gesture recognition performance using KLMS-RBFNN-GAF classifier with GA based optimized KM, DCT, proposed combined and proposed DWT-FR feature sets are 99.80%, 100.00%, 99.80% and 100.00%, respectively for Database I, 92.83%, 97.25%, 94.50% and 97.75%, respectively for Database II and 98.33%, 99.33%, 99.08% and 99.83%, respectively for Database III the average sensitivities of hand gesture recognition performance using KLMS-RBFNN-GAF classifier with original KM, DCT, proposed combined and proposed DWT-FR feature sets are 99.80%, 99.80%, 99.80% and 100.00%, respectively for Database I, 92.67%, 97.17%, 94.17% and 97.58%, respectively for Database II and 98.25%, 99.25%, 98.92% and 99.67%, respectively for Database III. Theses results indicate that GA based feature vector optimization technique with KLMS-RBFNN-GAF

Table 5.2: Performance of hand gesture recognition using KLMS-RBFNN-GAF classifier with original and GA based optimized KM, DCT, proposed combined and proposed DWT-FR feature sets for three different databases.

Database	Feature set	With original feature set				With GA based optimized feature set			
		Acc	Sen	Ppr	Spe	Acc	Sen	Ppr	Spe
Database I	KM	99.98	99.80	99.81	99.99	**99.98**	**99.80**	**99.81**	**99.99**
	DCT	99.98	99.80	99.81	99.99	**100.00**	**100.00**	**100.00**	**100.00**
	Combined	99.98	99.80	99.81	99.99	**99.98**	**99.80**	**99.81**	**99.99**
	DWT-FR	100.00	100.00	100.00	100.00	**100.00**	**100.00**	**100.00**	**100.00**
Database II	KM	99.39	92.67	92.73	99.68	**99.40**	**92.83**	**92.89**	**99.69**
	DCT	99.76	97.17	97.26	99.88	**99.77**	**97.25**	**97.32**	**99.88**
	Combined	99.51	94.17	94.26	99.75	**99.54**	**94.50**	**94.58**	**99.76**
	DWT-FR	99.80	97.58	97.64	99.89	**99.81**	**97.75**	**97.75**	**99.90**
Database III	KM	99.85	98.25	98.29	99.92	**99.86**	**98.33**	**98.35**	**99.93**
	DCT	99.94	99.25	99.27	99.97	**99.94**	**99.33**	**99.35**	**99.97**
	Combined	99.91	98.92	98.94	99.95	**99.92**	**99.08**	**99.10**	**99.96**
	DWT-FR	99.97	99.67	99.68	99.99	**99.99**	**99.83**	**99.84**	**99.99**

classifier improves the recognition performance in most of the cases except Database I. The ROC graphs of hand gesture recognition using proposed KLMS-RBFNN-GAF classifier with optimized KM, DCT, proposed combined and proposed DWT-FR feature sets for three distinct databases are shown in Figure 5.6.

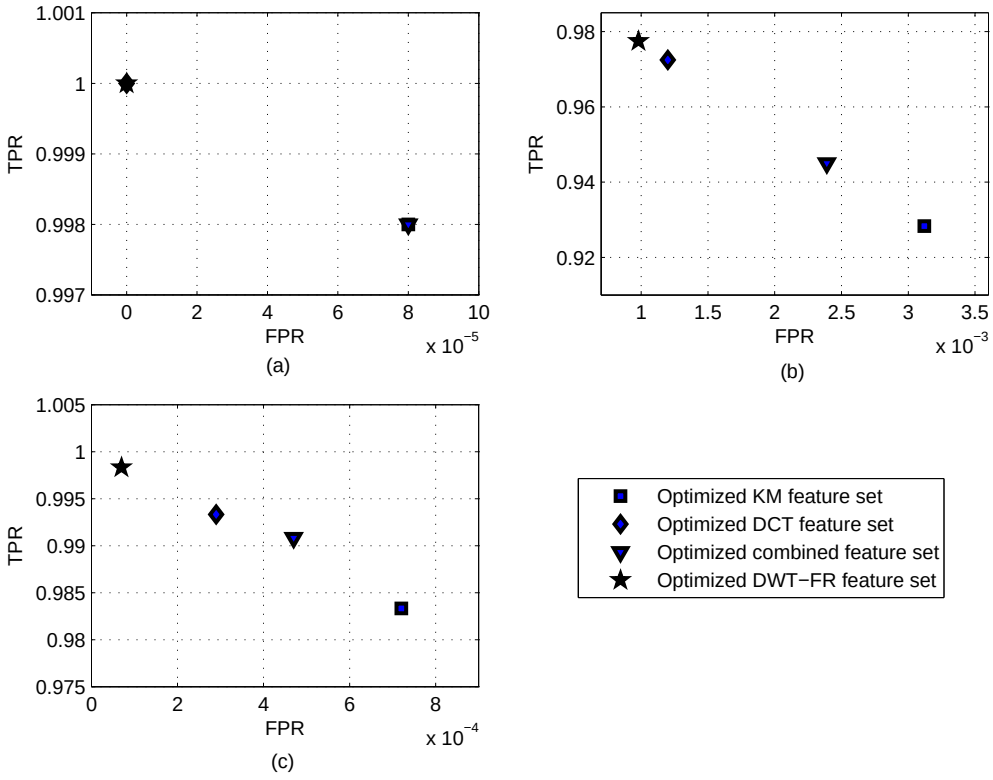


Figure 5.6: ROC graphs of hand gesture recognition using proposed KLMS-RBFNN-GAF classifier with optimized KM, DCT, proposed combined and proposed DWT-FR feature sets for (a) Database I, (b) Database II and (c) Database III.

5.4.2 Experimental Results using KLMS-RBFNN-SAF Classifier

The performance of the GA based feature vector optimization technique with KLMS-RBFNN-SAF classifier for KM, DCT, proposed combined and proposed DWT-FR feature sets are shown in Table 5.3. The results in Table 5.3 indicate that the lengths of GA based optimized KM, DCT, proposed combined and proposed DWT-FR feature sets

are 19, 38, 51 and 48, respectively for Database I, 45, 57, 181 and 127 respectively for Database II, and 34, 64, 200 and 83, respectively for Database III whereas the lengths of original KM, DCT, proposed combined and proposed DWT-FR feature sets are 64, 100, 163 and 192, respectively for Database I, 64, 100, 313 and 192, respectively for Database II and 64, 100, 421 and 192, respectively for Database III. Therefore, for KM, DCT, proposed combined and proposed DWT-FR feature sets, the proposed GA based feature vector optimization technique with KLMS-RBFNN-SAF classifier reduces number of features by 70.31%, 62.00%, 68.71% and 75.00%, respectively for Database I, 29.69%, 43.00%, 42.17% and 33.85%, respectively for Database II and 46.88%, 36.00%, 52.49% and 56.77%, respectively for Database III. From Table 5.1 and Table 5.3, it is noted that GA based feature vector optimization technique significantly reduces the dimension of feature vectors which in turn helps to reduce feature space complexity.

The performance of hand gesture recognition using KLMS-RBFNN-SAF classifier with original and GA based optimized KM, DCT, proposed combined and proposed DWT-FR feature sets for three different databases are shown in Table 5.4. From Table 5.4, it is observed that the average sensitivities of gesture recognition performance using KLMS-RBFNN-SAF classifier with GA based optimized KM, DCT, proposed combined and proposed DWT-FR feature sets are 99.80%, 100.00%, 99.80% and 100.00%, respectively for Database I, 93.33%, 97.58%, 94.67% and 98.50%, respectively for Database II and 98.42%, 99.42%, 99.33% and 99.83%, respectively for Database III. On the other hand, the average sensitivities of gesture recognition performance using KLMS-RBFNN-SAF classifier with original KM, DCT, proposed combined and proposed DWT-FR feature sets are 99.80%, 99.80%, 99.80% and 100.00%, respectively for Database I, 92.75%, 97.58%, 94.25% and 97.75%, respectively for Database II and 98.25%, 99.33%, 99.08% and 99.75%, respectively for Database III. The results indicate that the GA based feature vector optimization technique with KLMS-RBFNN-SAF classifier improves the recognition performance in most of the cases except Database I. Figure 5.7 shows the ROC graphs of hand gesture recognition using proposed KLMS-RBFNN-SAF classifier with optimized KM, DCT, proposed combined and proposed DWT-FR feature sets for three

Table 5.3: Performance of GA based feature vector optimization technique with KLMS-RBFNN-SAF classifier for KM, DCT, proposed combined and proposed DWT-FR feature sets.

Database	Feature set	Length of feature sets		% of reduction
		Original	Optimized	
Database I	KM	64	**19**	**70.31**
	DCT	100	**38**	**62.00**
	Combined	163	**51**	**68.71**
	DWT-FR	192	**48**	**75.00**
Database II	KM	64	**45**	**29.69**
	DCT	100	**57**	**43.00**
	Combined	313	**181**	**42.17**
	DWT-FR	192	**127**	**33.85**
Database III	KM	64	**34**	**46.88**
	DCT	100	**64**	**36.00**
	Combined	421	**200**	**52.49**
	DWT-FR	192	**83**	**56.77**

Table 5.4: Performance of hand gesture recognition using KLMS-RBFNN-SAF classifier with original and GA based optimized KM, DCT, proposed combined and proposed DWT-FR feature sets for three different databases.

Database	Feature set	With original feature set				With GA based optimized feature set			
		Acc	Sen	Ppr	Spe	Acc	Sen	Ppr	Spe
Database I	KM	99.98	99.80	99.81	99.99	**99.98**	**99.80**	**99.81**	**99.99**
	DCT	99.98	99.80	99.81	99.99	**100.00**	**100.00**	**100.00**	**100.00**
	Combined	99.98	99.80	99.81	99.99	**99.98**	**99.80**	**99.81**	**99.99**
	DWT-FR	100.00	100.00	100.00	100.00	**100.00**	**100.00**	**100.00**	**100.00**
Database II	KM	99.40	92.75	92.81	99.68	**99.44**	**93.33**	**93.39**	**99.71**
	DCT	99.80	97.58	97.64	99.89	**99.80**	**97.58**	**97.65**	**99.89**
	Combined	99.52	94.25	94.34	99.75	**99.56**	**94.67**	**94.80**	**99.77**
	DWT-FR	99.81	97.75	97.78	99.90	**99.88**	**98.50**	**98.52**	**99.93**
Database III	KM	99.85	98.25	98.28	99.92	**99.87**	**98.42**	**98.44**	**99.93**
	DCT	99.94	99.33	99.36	99.97	**99.95**	**99.42**	**99.42**	**99.97**
	Combined	99.92	99.08	99.11	99.96	**99.94**	**99.33**	**99.35**	**99.97**
	DWT-FR	99.98	99.75	99.76	99.99	**99.99**	**99.83**	**99.84**	**99.99**

different databases. Figure 5.8. From Table 5.2, Table 5.4, Figure 5.6 and Figure 5.7,

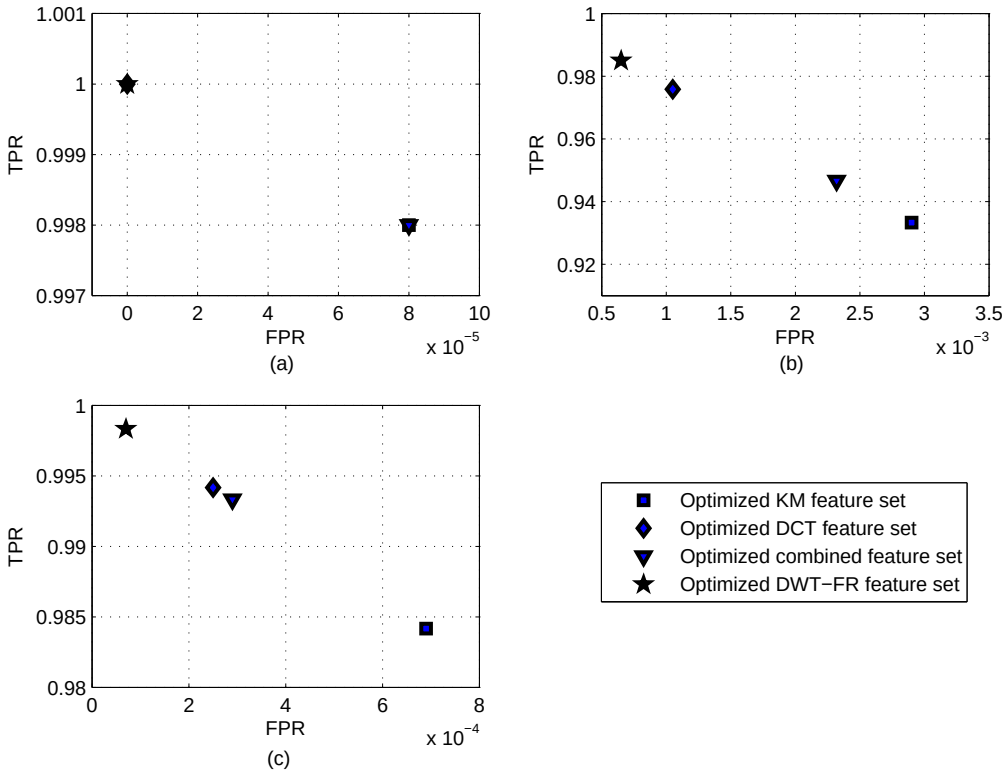


Figure 5.7: ROC graphs of hand gesture recognition using proposed KLMS-RBFNN-SAF classifier with optimized KM, DCT, proposed combined and proposed DWT-FR feature sets for (a) Database I, (b) Database II and (c) Database III.

it is observed that optimized proposed DWT-FR feature set offers the best recognition performance compared to all other feature sets used.

The comparative gesture recognition sensitivities using proposed KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers with optimized KM, DCT, proposed combined and proposed DWT-FR feature sets for three hand gesture databases are shown in From Figure 5.8, it is observed that with optimized KM, DCT, proposed combined and proposed DWT-FR feature sets, the recognition performance of proposed KLMS-RBFNN-SAF is better compared to KLMS-RBFNN-GAF classifier in most of the cases except Database I. In case of Database I, the KLMS-RBFNN-SAF classifier provides

an equal recognition sensitivity with KLMS-RBFNN-GAF classifier for optimized KM, DCT, proposed combined and proposed DWT-FR feature sets.

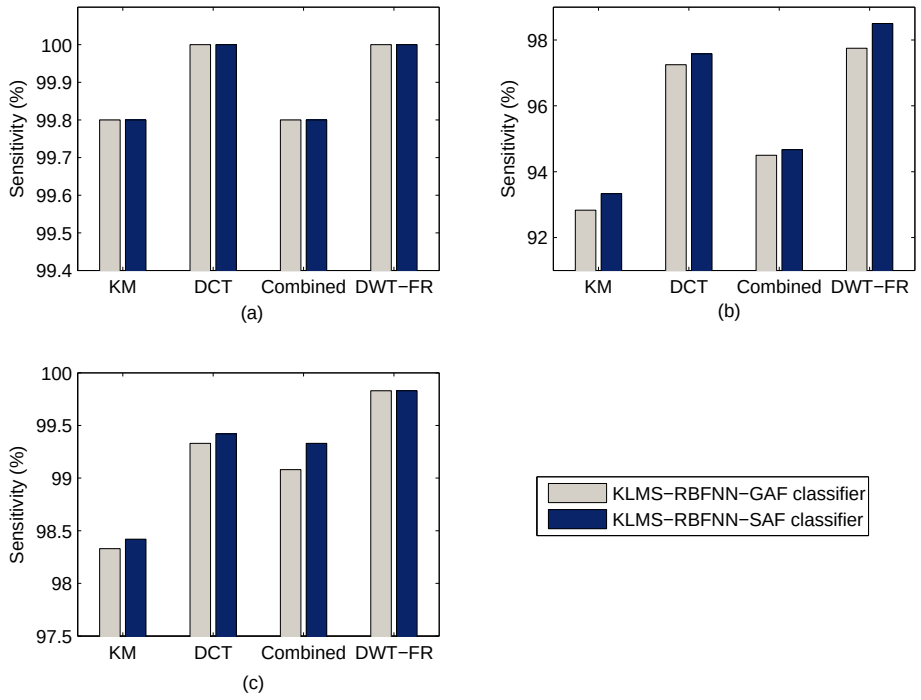


Figure 5.8: The average sensitivity of gesture recognition using proposed KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers with optimized KM, DCT, proposed combined and proposed DWT-FR feature sets for (a) Database I, (b) Database II and (c) Database III.

To evaluate the overall hand gesture recognition performance of proposed method1 (GA based optimized DWT-FR features with KLMS-RBFNN-GAF classifier represented as DWT-FR-GA-KLMS-RBFNN-GAF) and proposed method2 (GA based optimized proposed DWT-FR features with KLMS-RBFNN-SAF classifier represented as DWT-FR-GA-KLMS-RBFNN-SAF), and to compare them with other existing techniques such as Hu moment invariants with MLP-BP-NN classifier (HU-MLP-BP-NN) [13], complex moments with MLP-BP-NN classifier (CM-MLP-BP-NN) [86], normalized silhouette distance signal features with k-least absolute deviations classifier (NSDS-KLAD) [49], discrete cosine transform coefficients with k-nearest neighbors classifier (DCT-KNN) [51]

and Krawtchouk moments with minimum distant classifier (KM-MD) [5], the experiments are conducted separately on Databases I, II and III with 2-fold cross validation test. The advantages of cross validation test are: (i) test sets are independent and (ii) the reliability of the performance results could be improved. For 2-fold cross validation test, all of the gesture images in the Databases I, II and III are divided into two equal parts. At each instance one part is taken for test dataset and remaining part is used as train dataset. The overall gesture recognition performance of proposed method1, proposed method2 and other existing techniques like HU-MLP-BP-NN [13], CM-MLP-BP-NN [86], NSDS-KLAD [49], DCT-KNN [51] and KM-MD [5] for Databases I, II and III are shown in Table 5.5.

From Table 5.5, it is observed that overall hand gesture recognition accuracy, sensitivity, positive predictivity and specificity of proposed method1 are 99.98%, 99.70%, 99.71% and 99.99%, respectively for Database I, 99.79%, 97.46%, 97.48% and 99.89%, respectively for Database II and 99.96%, 99.50%, 99.51% and 99.98%, respectively for Database III whereas overall gesture recognition accuracy, sensitivity, positive predictivity and specificity of proposed method2 are 99.98%, 99.80%, 99.80% and 99.99%, respectively for Database I, 99.80%, 97.63%, 97.64% and 99.90%, respectively for Database II and 99.98%, 99.75%, 99.75% and 99.99%, respectively for Database III.

On the other hand, HU-MLP-BP-NN method [13] achieves accuracy, sensitivity, positive predictivity and specificity as 98.01%, 75.10%, 71.34% and 98.96%, respectively for Database I, 94.91%, 38.88%, 35.76% and 97.34%, respectively for Database II and 95.55%, 46.54%, 40.71% and 97.68%, respectively for Database III, CM-MLP-BP-NN method [86] shows accuracy, sensitivity, positive predictivity and specificity as 99.42%, 92.70%, 92.95% and 99.70%, respectively for Database I, 97.37%, 68.42%, 67.87% and 98.63%, respectively for Database II and 98.33%, 80.00%, 80.37% and 99.13%, respectively for Database III, NSDS-KLAD method [49] provides accuracy, sensitivity, positive predictivity and specificity as 99.68%, 96.00%, 96.35% and 99.83%, respectively for Database I, 98.93%, 87.17%, 87.30% and 99.44%, respectively for Database II and 99.60%, 95.17%, 95.26% and 99.79%, respectively for Database III. The DCT-KNN

Table 5.5: The comparative performances of different hand gesture recognition techniques for Databases I, II and III.

Database	Method	Acc (%)	Sen (%)	Ppr (%)	Spe (%)
Database I	HU-MLP-BP-NN [13]	98.01	75.10	71.34	98.96
	CM-MLP-BP-NN [86]	99.42	92.70	92.95	99.70
	NSDS-KLAD [49]	99.68	96.00	96.35	99.83
	DCT-KNN [51]	99.90	98.80	98.90	99.95
	KM-MD [5]	99.93	99.10	99.14	99.96
	Proposed method1	**99.98**	**99.70**	**99.71**	**99.99**
	Proposed method2	**99.98**	**99.80**	**99.80**	**99.99**
Database II	HU-MLP-BP-NN [13]	94.91	38.88	35.76	97.34
	CM-MLP-BP-NN [86]	97.37	68.42	67.87	98.63
	NSDS-KLAD [49]	98.93	87.17	87.30	99.44
	DCT-KNN [51]	99.20	90.46	91.14	99.59
	KM-MD [5]	99.26	91.13	91.34	99.61
	Proposed method1	**99.79**	**97.46**	**97.48**	**99.89**
	Proposed method2	**99.80**	**97.63**	**97.64**	**99.90**
Database III	HU-MLP-BP-NN [13]	95.55	46.54	40.71	97.68
	CM-MLP-BP-NN [86]	98.33	80.00	80.37	99.13
	NSDS-KLAD [49]	99.60	95.17	95.26	99.79
	DCT-KNN [51]	99.59	95.13	95.34	99.79
	KM-MD [5]	99.77	97.29	97.31	99.88
	Proposed method1	**99.96**	**99.50**	**99.51**	**99.98**
	Proposed method2	**99.98**	**99.75**	**99.75**	**99.99**

method [51] yields accuracy, sensitivity, positive predictivity and specificity as 99.90%, 98.80%, 98.90% and 99.95%, respectively for Database I, 99.20%, 90.46%, 91.14% and 99.59%, respectively for Database II and 99.59%, 95.13%, 95.34% and 99.79%, respectively for Database III, and KM-MD method [5] provides accuracy, sensitivity, positive predictivity and specificity as 99.93%, 99.10%, 99.14% and 99.96%, respectively for Database I, 99.26%, 91.13%, 91.34% and 99.61%, respectively for Database II and 99.77%, 97.29%, 97.31% and 99.88%, respectively for Database III. From the comparative results, it is noted that HU-MLP-BP-NN method [13] provides least performance compared to all other methods to recognize static hand gesture images, because it uses 7 Hu moment invariants as a feature set which is unable to distinguish large number of gestures having similar shapes. From Table 5.5, it is also observed that the proposed methods provide a significantly higher recognition performance compared to other existing methods such as HU-MLP-BP [13], CM-MLP-BP [86], NSDS-KLAD [49], DCT-KNN [51] and KM-MD [5].

The proposed methods achieve higher recognition performance due to following reasons: (i) they use homomorphic filtering [67] and gray world [68] techniques to compensate the lighting variation of grayscale and color gesture images respectively; (ii) to segment hand region, this work uses histogram based Otsu segmentation algorithm for grayscale image and skin color detection based color segmentation technique for color image; (iii) the proposed image rotation technique makes the image as rotation-invariant; (iv) a position and size invariant proposed DWT and F-ratio based feature set which carries localized time and spatial frequency information of the hand gesture image, is used for gesture representation; (v) GA based optimized feature subset selection technique is used here to select more important and discriminative feature by removing unwanted features from proposed DWT and F-ratio based feature set; (vi) the modified RBF neural network known as KLMS-RBFNN-GAF or KLMS-RBFNN-SAF is used here to classify hand gesture images.

For the results given in Table 5.5, the gesture-wise overall recognition performance of the proposed method1 and proposed method2 for Database II are shown in Table 5.6 and

Table 5.7, respectively. From the results in Table 5.6 and Table 5.7, it is observed that

Table 5.6: Confusion matrix for the performance of overall gesture recognition using proposed method1 as shown in Table 5.5 for Database II.

Gesture		O/P																						
		A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
I/P	A	**100**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	B	0	**100**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	C	0	1	**99**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	D	0	0	0	**100**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	E	0	0	2	0	**96**	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0
	F	0	0	1	0	0	**98**	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	G	0	0	0	0	0	0	**100**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	H	0	0	0	0	0	0	0	**100**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	I	0	0	0	0	0	0	0	0	**100**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	K	0	0	0	0	0	0	0	0	0	**100**	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	L	0	0	0	0	0	0	0	0	0	0	**100**	0	0	0	0	0	0	0	0	0	0	0	0	0
	M	0	0	0	0	1	0	0	0	0	0	0	**94**	3	0	0	0	0	1	1	0	0	0	0	0
	N	0	0	0	0	4	0	0	0	0	0	0	3	**91**	0	0	0	0	0	2	0	0	0	0	0
	O	0	0	0	0	1	0	0	0	0	0	0	0	0	**98**	0	0	0	0	0	0	0	0	1	0
	P	0	0	0	0	0	0	0	1	0	0	0	0	0	0	**98**	0	1	0	0	0	0	0	0	0
	Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**98**	1	0	0	0	0	0	0	1
	R	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	**99**	0	0	0	0	0	0	0
	S	2	1	0	0	2	0	0	0	0	0	0	3	4	1	0	0	0	**86**	1	0	0	0	0	0
	T	2	0	0	0	0	0	0	0	0	0	0	1	3	0	0	0	0	1	**93**	0	0	0	0	0
	U	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**100**	0	0	0	0
	V	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	**99**	0	0	0
	W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**100**	0	0
	X	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	**97**	0
	Y	0	0	0	1	0	0	0	1	2	0	2	0	0	0	1	0	0	0	0	0	0	0	0	**93**

the individual gesture recognition performance of the proposed methods for few gestures such as 'E', 'M', 'N', 'S', 'T', etc. are comparably lower. This is because the shape and contrasting homogeneous regions of these gestures are more similar and overlap with other gestures. However, the overall static hand gesture recognition performance of the proposed methods are better compared to earlier reported techniques.

The proposed static gesture recognition methods are implemented using MATLAB 7.6.0 (R2008a). When executed on a Intel Core i5 computer, processor 3.20Ghz with 4GB RAM, Windows 7 platform and as only application running, proposed methods take an average execution time approximately 0.39 seconds for recognition of each test gesture image. Note that Databases II and III both are constructed incorporating the variation of position, rotation and size of the gesture images under different illumination conditions. Therefore, all the recognition performances provided in the above experimental study are obtained under the variation of position, rotation and size of the gesture images along with different illumination conditions.

Table 5.7: Confusion matrix for the performance of overall gesture recognition using proposed method2 as shown in Table 5.5 for Database II.

Gesture		O/P																						
		A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
I/P	A	**100**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	B	0	**100**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	C	0	0	**100**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	D	0	0	0	**98**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
	E	0	0	2	0	**95**	0	0	0	0	0	0	2	0	0	0	0	0	1	0	0	0	0	0	0
	F	0	1	1	0	0	**98**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	G	0	0	0	0	0	0	**100**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	H	0	0	0	0	0	0	0	**99**	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
	I	0	1	0	0	0	0	0	0	**98**	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	K	0	0	0	0	0	0	0	0	0	**100**	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	L	0	0	0	0	0	0	0	0	0	0	**100**	0	0	0	0	0	0	0	0	0	0	0	0	0
	M	0	1	0	0	0	0	0	0	0	0	0	**93**	1	0	0	0	0	4	1	0	0	0	0	0
	N	0	0	0	0	0	0	0	0	0	0	0	3	**94**	0	0	0	0	1	1	1	0	0	0	0
	O	0	0	0	0	0	0	0	0	0	0	0	1	0	**99**	0	0	0	0	0	0	0	0	0	0
	P	1	0	0	0	0	0	0	2	0	0	0	0	0	0	**97**	0	0	0	0	0	0	0	0	0
	Q	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	**98**	0	0	0	0	0	0	0	1
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**100**	0	0	0	0	0	0	0
	S	0	1	0	0	1	0	0	0	0	0	0	2	0	0	0	0	0	**92**	4	0	0	0	0	0
	T	1	0	0	0	1	0	0	0	0	0	0	1	5	0	0	0	0	1	**91**	0	0	0	0	0
	U	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	**99**	0	0	0	0
	V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**100**	0	0	0
	W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**100**	0	0
	X	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	**98**	0
	Y	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	1	1	**94**

5.5 Conclusions

The conclusions of the chapter are presented as follows.

- In this chapter, a GA based feature vector optimization technique is adopted to optimize the feature set for hand gesture recognition. The objective of feature vector optimization is to select a subset of features that contain the most informative and discriminative features. The lengths of the extracted feature sets are reduced by eliminating the redundant and irrelevant features using GA based feature vector optimization technique with proposed KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers.

- The experiments are separately conducted on three hand gesture databases using KM, DCT, proposed combined and proposed DWT-FR feature sets to evaluate the performance of GA based feature vector optimization technique. Experimental results indicate that GA based feature vector optimization technique reduce a significant number of features which in turn helps to reduce complexity of the proposed hand gesture recognition technique. Experimental results also indicate that in most of the cases, the proposed feature vector optimization technique

improves the recognition performance for both KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers. From experimental study, it is also observed that the KLMS-RBFNN-SAF classifier with GA based optimized DWT-FR features provides best recognition performance for all three databases.

- To evaluate the overall recognition performance of proposed method1 (DWT-FR-GA-KLMS-RBFNN-GAF) and proposed method2 (DWT-FR-GA-KLMS-RBFNN-GAF) and to compare with other existing techniques, the experiments are conducted separately on three databases with 2-fold cross validation test. The comparative results indicate that the proposed methods provide significantly higher recognition performance compared to other existing methods such as HU-MLP-BP-NN [13], CM-MLP-BP-NN [86], NSDS-KLAD [49], DCT-KNN [51] and KM-MD [5].

CHAPTER 6

Conclusions and Future Work

6.1 Conclusions

This thesis is embodied with the results of our investigation on preprocessing, feature extraction, feature vector optimization and classification stages of vision-based static hand gesture recognition technique. In particular, our investigations have been addressed towards improving the performance of a vision-based hand gesture recognition system which overcomes the challenges of illumination, rotation, size and position variation of the gesture images.

- In Chapter 2, the preprocessing stage of vision-based hand gesture recognition technique has been described. The preprocessing stage contains the following substages: image enhancement which enhances the image by reducing illumination variation; segmentation, which segments hand region from its background image and transforms into binary silhouette; image rotation that makes segmented gesture rotation invariant; filtering that effectively removes background noise and object noise from binary image and provides a well defined segmented hand gesture image. This chapter proposes a novel image rotation technique that makes the segmented gesture as rotation invariant by coinciding the first principal component of the segmented hand gestures with vertical axes. This chapter also proposes a combined feature set by appending LCS features with block-based features for better representation of hand gesture images. In feature extraction stage, two types of features are extracted from preprocessed hand gesture image: (i) Localized contour sequences (LCS): a contour-based feature set that convey the contour information of the hand gesture images. (ii) Block-based features: the regional information of the hand gesture image is incorporated in this features. Finally, a combined feature set which couples LCS features with block-based features is proposed to represent the static hand gesture image. The proposed combined feature set carries the contour as well as regional information of the gesture and provides a better representation of hand gesture image compared to each of the individual LCS or block-based feature set. This combined feature set is applied to the input of the

classifier to recognize the class of static hand gesture images. The performance of gesture recognition is tested on three hand gesture databases which include grayscale images with uniform background (Databases I and II) and color images with non uniform background (Database III). Database I consists of 1000 static gesture images of 25 D/ISL hand alphabets. On the other hand, both Databases II and III contain 2400 static gesture images of 24 ASL hand alphabets. To investigate the performance of the proposed combined feature set and compare with LCS and block-based feature sets, the experiments are separately conducted using LCS, block-based and proposed combined feature sets with MLP-NN classifier. Experimental results indicate that the proposed combined feature set provides better recognition performance compared to LCS [16] and block-based [17] feature sets individually. Experimental results also demonstrate that gesture recognition performances are significantly improved due to the proposed image rotation technique.

- In chapter 3, a feature extraction technique to extract the DWT and F-ratio based feature set for more efficient representation of hand gesture image. In the proposed feature extraction technique, the Haar wavelet is applied on the resized and enhanced grayscale image, and then the important DWT coefficient matrices are selected as features using the proposed F-ratio based technique to represent each hand gesture image. The proposed DWT and F-ratio based (DWT-FR) feature set offers more efficient representation of hand gesture image because it enjoys following advantages: (i) this work uses Haar wavelet which is very simple, effectively describes the soft parts of human body and follow the orthonormal property of wavelet basis; (ii) the DWT offers a simultaneous localization information in time and spatial frequency domains; (iii) the DWT provides multi-resolution analysis of an image using higher level decomposition; and (iv) the best DWT coefficient matrices are selected as features using an F-ratio based technique. The proposed DWT and F-ratio based feature set is applied as input to the RBF neural network (RBF-NN) to recognize the static hand gesture images. This chapter

introduces RBF-NN as classifier because it has following advantages: (i) its architecture is very simple, only one hidden layer consists between input and output layers; (ii) in hidden layer, localized radial basis functions are used to perform nonlinear transform of input vector from input space to higher dimensional hidden space where vector are more linearly separable; (iii) sensitivity of the hidden neuron is tuned by adjusting parameters of basis function; (iv) this network is free from local minima problem. To verify gesture recognition performance of the proposed DWT-FR feature set and to compare it with proposed combined feature set shown in chapter 1 and other earlier reported techniques such as KM [5] and DCT [51] feature sets, the experiments are separately conducted on three hand gesture databases using both MLP-BP-NN and RBF-NN classifiers. Experimental results demonstrate that the proposed DWT-FR feature set provides better recognition performance compared to KM [5], DCT [51] and proposed combined features for all three databases. Experimental results also demonstrate that RBF-NN classifier achieves higher performance for hand gesture recognition compared to MLP-BP-NN classifier.

- In chapter 4, a modified RBF-NN classifier which is k-mean and LMS based RBF-NN with Gaussian activation functions (represented as KLMS-RBFNN-GAF classifier) is proposed for better recognition of hand gesture images. In KLMS-RBFNN-GAF classifier, the Gaussian function is used as activation function, the centers are automatically selected through k-means based center-selection algorithm and the estimated weight matrix is further updated using least mean square (LMS) algorithm. The aim of updation technique is to update the estimated weight matrix at the training phase such that the training error will be minimized. The training error represents the error between score generated output and actual target of the network at the training phase. To implement this idea, LMS algorithm is adopted to update the estimated weight matrix in training phase such that it minimizes the mean square error (MSE). This in turn helps to improve the gesture recognition performance. However, one of the difficulties

in RBF-NN is the use of Gaussian function which is inefficient to estimate the similar patterns. Therefore, this work also proposes a sigmoidal function based modified RBF-NN classifier i.e. k-mean and LMS-based RBF-NN classifier with composite sigmoidal activation function (represented as KLMS-RBFNN-SAF classifier). In KLMS-RBFNN-SAF classifier, the activation function is formed using a set of composite sigmoidal functions, centers are automatically selected using k-mean algorithm and estimated weight matrix is further updated using LMS algorithm. The aim of the proposed KLMS-RBFNN-SAF classifier is to improve the recognition performance with the help of better estimation of the similar training patterns. For both of proposed classifiers, selected centers and updated weight matrix are stored at end of the training phase and these are used to recognize hand gesture images at testing phase. The experiments are separately conducted on three hand gesture databases using KM [5], DCT [51], proposed combined and proposed DWT-FR feature sets to evaluate gesture recognition performance of the proposed KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers and to compare them with other reported classifiers such as MLP-BP-NN and RBF-NN. Experimental results indicate that the proposed modified RBF-NN classifiers provide better recognition performance compared to other reported classifiers such as MLP-BP-NN and RBF-NN. Experimental results also indicate that the KLMS-RBFNN-SAF classifier offers better or equal recognition performance compared to KLMS-RBFNN-GAF classifier. From experimental results, it is also observed that the performance of hand gesture recognition using DWT-FR features is better compared to using KM [5], DCT [51] and proposed combined features for all different classifiers including MLP-BP-NN, RBF-NN, KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers.

- In Chapter 5, a GA based feature vector optimization technique is adopted to optimize the original feature set for hand gesture recognition. The objective of the feature vector optimization is to reduce the dimension of feature vector by selecting an optimal subset of features that contains the most informative and

discriminative information of patterns. The reduction of feature dimensionality contributes several advantages for a recognition system: (i) reduce the length of feature vector which in turn helps to reduce feature dimensional complexity and the cost of acquisition of the data, (ii) increase the understandability of the final classification model, (iii) speed up the final classification mode, (iv) an improvement in classification performance. The length of extracted feature sets are reduced in this work by eliminating the redundant and irrelevant features using GA with proposed KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers. To evaluate the performance of GA based feature vector optimization technique, the experiments are separately conducted on three hand gesture databases using KM, DCT, proposed combined and proposed DWT-FR feature sets. Experimental results indicate that GA based feature vector optimization techniques reduces a significant number of features which in turn helps to reduce complexity of the proposed hand gesture recognition technique. Experimental results also indicate that in most of the cases, GA based feature vector optimization technique improves the hand gesture recognition performance for both KLMS-RBFNN-GAF and KLMS-RBFNN-SAF classifiers. From experimental study it is also observed that the KLMS-RBFNN-SAF classifier with GA based optimized DWT-FR feature set provides the best recognition performance for all three databases. To compare the overall hand gesture recognition performance of proposed method1 (GA based optimized DWT-FR features with KLMS-RBFNN-GAF classifier represented as DWT-FR-GA-KLMS-RBFNN-GAF) and proposed method2 (GA based optimized DWT-FR features with KLMS-RBFNN-SAF classifier represented as DWT-FR-GA-KLMS-RBFNN-SAF) and compare with other existing techniques such as Hu moment invariants with MLP-BP-NN classifier (HU-MLP-BP-NN) [13], complex moments with MLP-BP-NN classifier (CM-MLP-BP-NN) [86], normalized silhouette distance signal features with k-least absolute deviations classifier (NSDS-KLAD) [49], discrete cosine transform coefficients with k-nearest neighbors classifier (DCT-KNN) [51] and Krawtchouk moments with minimum distant classifier

(KM-MD) [5], the experiments are conducted separately on three hand gesture databases with 2-fold cross validation test. The comparative results indicate that the proposed methods (DWT-FR-GA-KLMS-RBFNN-GAF and DWT-FR-GA-KLMS-RBFNN-SAF) provide significantly higher recognition performance compared to other existing methods such as HU-MLP-BP-NN [13], CM-MLP-BP-NN [86], NSDS-KLAD [49], DCT-KNN [51] and KM-MD [5].

- Finally, all the results presented throughout the thesis demonstrate the superiority of our different propositions. We believe that the proposed method would also maintain the consistency of the results if some other hand gesture data set is used.

6.2 Future Research Directions

In this section, a variety of research directions, future scope and possible extensions of this thesis work are discussed.

- Although the proposed features perform significantly well for static hand gesture recognition, it is observed that the proposed features are unable to properly classify few gestures such as 'E', 'M', 'N', 'S', 'T' etc. The shape of these gestures are similar and overlap to each other. Therefore, one could think of a complementary feature set that can be used for properly distinguishing the gestures having similar shape.

- In order to improve the recognition performance, an appropriate feature transformation technique may be applied to the extracted feature vectors. In transform domain, the feature vector should contain decorrelated and low dimension features and should be more linearly classifiable.

- A significant amount of variations in hand gesture images among different users sometimes make misclassification during recognition. The development of an unsupervised classifier will be a possible solution for recognition of hand gesture images. This is because an unsupervised classifier neither requires fore-knowledge

of the gesture classes, nor demands any training dataset to recognize hand gesture images.

- The static hand gestures recognition system frameworks developed in this thesis are tested in off-line. However, the system frameworks may be extended for on-line testing so that they can be used in various real-time applications such as human-robot interaction, medical systems, TV production, video editing, and sign language recognition.

- One of the interesting extension of this work can be tracking and recognizing dynamic and two hands gestures. Two hands tracking can be integrated with virtual environments to communicate with the computer and objects in the virtual environment in a more natural way. The more advanced extension of this work can be hand tracking in a 3D space.

- Another possible interesting extension of this work is the development of multi-modal tracking and recognition method where multiple objects such as hand gesture, face expression, eye gaze and voice can be tracked and recognized at the same time. The multi-modal tracking and recognition strategy will improve the performance of human computer interaction.

References

[1] S. Mitra and T. Acharya, "Gesture Recognition: A Survey," *IEEE Trans. Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 3, pp. 311 –324, May. 2007.

[2] D. K. Ghosh and S. Ari, "A Static Hand Gesture Recognition Algorithm using K-Mean based Radial Basis Function Neural Network," in *Proc. 8th Int. Conf. Information, Communications, and Signal Processing (ICICS)*, Singapore, Dec. 2011, pp. 1–5.

[3] M. R. Lehto, S. J. Landry, and J. Buck, *Introduction to Human Factors and Ergonomics for Engineers*. CRC Press, 2007.

[4] E. Gingir, "Hand Gesture Recognition System," Master's thesis, Middle East Technical University, 2010.

[5] S. Padam Priyal and P. K. Bora, "A Robust Static Hand Gesture Recognition System Using Geometry Based Normalizations and Krawtchouk Moments," *Pattern Recogn.*, vol. 46, no. 8, pp. 2202–2219, Aug. 2013.

[6] C. Manresa, J. Varona, R. Mas, and F. J. Perales, "Hand Tracking and Gesture Recognition for Human-Computer Interaction," *Electronic Letters on Computer Vision and Image Analysis*, vol. 5, no. 3, pp. 96–104, 2005.

[7] H. Hasan and S. Abdul-Kareem, "Static Hand Gesture Recognition Using Neural Networks," *Artif. Intell. Rev.*, vol. 41, no. 2, pp. 147–181, Feb. 2014.

[8] S. C. W. Ong and S. Ranganath, "Automatic sign language analysis: a survey and the future beyond lexical meaning," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 873 – 891, Jun. 2005.

[9] D. K. Ghosh and S. Ari, "On an algorithm for Vision-based hand gesture recognition," *Signal, Image and Video Processing*, vol. 10, no. 4, pp. 1–8, Apr. 2016.

[10] H. Badi and S. Hussein, "Hand posture and gesture recognition technology," *Neural Computing and Applications*, vol. 25, no. 3-4, pp. 871–878, Sep. 2014.

[11] R. M. Krauss, R. A. Dushay, Y. Chen, and F. Rauscher, "The Communicative Value of Conversational Hand Gesture," *Journal of Experimental Social Psychology*, vol. 31, no. 6, pp. 533–552, Nov. 1995.

[12] O. Al-Jarrah and A. Halawani, "Recognition of gestures in Arabic sign language using neuro-fuzzy systems," *Artif. Intell.*, vol. 133, no. 1-2, pp. 117–138, Dec. 2001.

[13] P. Premaratne and Q. Nguyen, "Consumer electronics control system based on hand gesture moment invariants," *IET Computer Vision*, vol. 1, no. 1, pp. 35 –41, Mar. 2007.

[14] F. Erden and A. E. Çetin, "Hand gesture based remote control system using infrared sensors and a camera," *IEEE Transactions on Consumer Electronics*, vol. 60, no. 4, pp. 675–680, Nov 2014.

[15] C. L. Nehaniv, K. Dautenhahn, J. Kubacki, M. Haegele, C. Parlitz, and R. Alami, "A methodological approach relating the classification of gesture to identification of human intent in the context of human-robot interaction," in *IEEE Int. Works. Robot and Human Interactive Communication, ROMAN 2005*, Aug. 2005, pp. 371–377.

[16] L. Gupta and S. Ma, "Gesture-based interaction and communication: automated classification of hand gesture contours," *IEEE Trans. Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 31, no. 1, pp. 114 –120, Feb. 2001.

[17] J. P. Wachs, H. Stern, and Y. Edan, "Cluster labeling and parameter estimation for the automated setup of a hand-gesture recognition system," *IEEE Trans. Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 35, no. 6, pp. 932 – 944, Nov. 2005.

[18] S. Bourennane and C. Fossati, "Comparison of shape descriptors for hand posture recognition in video," *Signal, Image and Video Processing*, vol. 6, no. 1, pp. 147–157, Mar. 2012.

[19] R. Yang, S. Sarkar, and B. Loeding, "Handling Movement Epenthesis and Hand Segmentation Ambiguities in Continuous Sign Language Recognition Using Nested Dynamic Programming," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 3, pp. 462–477, Mar. 2010.

[20] M. K. Bhuyan, P. K. Bora, and D. Ghosh, "An Integrated Approach to the Recognition of a Wide Class of Continuous Hand Gestures," *Int. J. Patt. Recogn. Artif. Intell.*, vol. 25, no. 2, pp. 227–252, Mar. 2011.

[21] X. Shen, G. Hua, L. Williams, and Y. Wu, "Dynamic Hand Gesture Recognition: An Exemplar-based Approach from Motion Divergence Fields," *Image Vision Comput.*, vol. 30, no. 3, pp. 227–235, Mar. 2012.

[22] W. Liu, Y. Fan, Z. Li, and Z. Zhang, "RGBD video based human hand trajectory tracking and gesture recognition system," *Mathematical Problems in Engineering*, vol. 2015, no. 863732, pp. 1–15, 2015.

[23] C. C. Chang, J. J. Chen, W. K. Tai, and C. C. Han, "New Approach for Static Gesture Recognition," *J. Inf. Sci. Eng.*, vol. 22, no. 5, pp. 1047–1057, Jan. 2006.

[24] D. J. Sturman and D. Zeltzer, "A survey of glove-based input," *IEEE Computer Graphics and Applications*, vol. 14, no. 1, pp. 30 –39, Jan. 1994.

[25] C. Wang and D. J. Cannon, "A virtual end-effector pointing system in point-and-direct robotics for inspection of surface flaws using a neural network based skeleton transform," in *IEEE Int. Conf. Robotics and Automation*, vol. 3, May. 1993, pp. 784–789.

[26] F. Parvini, D. McLeod, C. Shahabi, B. Navai, B. Zali, and S. Ghandeharizadeh, "An Approach to Glove-Based Gesture Recognition," in *Human-Computer Interaction. Novel Interaction Methods and Techniques, Lecture Notes in Computer Science*, J. A. Jacko, Ed. Springer Berlin Heidelberg, 2009, vol. 5611, pp. 236–245.

[27] M. E. Cabrera, J. M. Bogado, L. Fermin, R. Acuna, and D. Ralev, "Glove-based gesture recognition system," in *Proc. 5th int. conf. climbing and walking robots and the support technologies for mobile machines, CLAWAR 2012*. Baltimore, USA, July 2012, pp. 747–753.

[28] G. Simion, V. Gui, and M. Otesteanu, "Vision Based Hand Gesture Recognition: A Review," *International journal of circuits, systems and signal processing*, vol. 6, no. 4, pp. 275–282, 2012.

[29] J.-T. Xue, Y.-R. Zong, and H.-W. Li, "A visual-based research on static gesture recognition," in *Int. Conf. Machine Learning and Cybernetics*, vol. 01, July 2013, pp. 476–480.

[30] S. Genç, M. Bastan, U. Güdükbay, V. Atalay, and Ö. Ulusoy, "HandVR: a hand-gesture-based interface to a video retrieval system," *Signal, Image and Video Processing*, vol. 9, no. 7, pp. 1717–1726, 2015.

[31] T. N. Nguyen, D. H. Vo, H. H. Huynh, and J. Meunier, "Geometry-based static hand gesture recognition using support vector machine," in *13th Int. Conf. Control Automation Robotics Vision (ICARCV), 2014*, Dec. 2014, pp. 769–774.

[32] G. Simion, V. Gui, and M. Otesteanu, "Vision Based Hand Gesture Recognition: A Review," *Iinternational journal of circuits, systems and signal processing*, vol. 6, no. 1, pp. 275–282, 2012.

[33] S. Meena, "A Study on Hand Gesture Recognition Technique," Master's thesis, National Institute of Technology Rourkela, India, 2011.

[34] A. Katkere, E. Hunter, D. Kuramura, J. Schlenzig, S. Moezzi, and R. Jain, "Robogest: Telepresence using hand gestures," Tech. Rep. VCL-94-104, Visual Computing Laboratory, University of California, San Diego, December 1994.

[35] M. Becker, E. Kefalea, E. Maël, C. Von Der Malsburg, M. Pagel, J. Triesch, J. Vorbrüggen, R. Würtz, and S. Zadel, "GripSee: A Gesture-Controlled Robot for Object Perception and Manipulation," *Autonomous Robots*, vol. 6, no. 2, pp. 203–221, Apr. 1999.

[36] A. Malima, E. Ozgur, and M. Cetin, "A Fast Algorithm for Vision-Based Hand Gesture Recognition for Robot Control," in *14th IEEE conf. Signal Processing and Communications Applications*, April 2006, pp. 1–4.

[37] H. Grant and C.-K. Lai, "Simulation modeling with artificial reality technology (SMART): an integration of virtual reality and simulation modeling," in *Simulation Conference Proceedings*, vol. 1, Dec. 1998, pp. 437–441.

[38] G. R. S. Murthy and R. S. Jadon, "A Review of Vision Based Hand Gestures Recognition," *International Journal of Information Technology and Knowledge Management*, vol. 2, no. 2, pp. 405–410, 2009.

[39] R. Sharma, T. Huang, Y. Zhao, and S. Chu, "Speech/gesture interface to a visual computing environment for molecular biologists," in *Proc. 13th Int. Conf. Pattern Recognition,*, vol. 3, Aug. 1996, pp. 964–968.

[40] G. Dong, Y. Yan, and M. Xie, "Vision-based hand gesture recognition for human-vehicle interaction," in *Proc. Int. conf. Control, Automation and Computer Vision*, vol. 1, 1998, pp. 151–155.

[41] C. A. Pickering, K. J. Burnham, and M. J. Richardson, "A Research Study of Hand Gesture Recognition Technologies and Applications for Human Vehicle Interaction," in *3rd IET Conf. Automotive Electronics*, June 2007, pp. 1–15.

[42] Q. Munib, M. Habeeb, B. Takruri, and H. A. Al-Malik, "American sign language (ASL) recognition based on Hough transform and neural networks," *Expert Syst. Appl.*, vol. 32, no. 1, pp. 24–37, 2007.

[43] X. Teng, B. Wu, W. Yu, and C. Liu, "A hand gesture recognition system based on local linear embedding," *J. Vis. Lang. Comput.*, vol. 16, no. 5, pp. 442–454, Oct. 2005.

[44] D. Stotts, J. M. Smith, and K. Gyllstrom, "FaceSpace: Endo- and Exo-spatial Hypermedia in the Transparent Video Facetop," in *Proc. 15th ACM Conf. Hypertext and Hypermedia.* New York, NY, USA: ACM, 2004, pp. 48–57.

[45] G. M. Smith and m. c. schraefel, "The Radial Scroll Tool: Scrolling Support for Stylus- or Touch-based Document Navigation," in *Proc. 17th Annual ACM Symp. User Interface Software and Technology, UIST '04.* New York, NY, USA: ACM, 2004, pp. 53–56.

[46] J. P. Wachs, H. I. Stern, Y. Edan, M. Gillam, J. Handler, C. Feied, and M. Smith, "A gesture-based tool for sterile browsing of radiology images," *Journal of the American Medical Informatics Association*, vol. 15, no. 3, pp. 321–323, 2008.

[47] S. P. Priyal and P. K. Bora, "A study on static hand gesture recognition using moments," in *Int. Conf. Signal Processing and Communications (SPCOM)*, Jul. 2010, pp. 1–5.

[48] Q. Chen, N. D. Georganas, and E. M. Petriu, "Hand Gesture Recognition Using Haar-Like Features and a Stochastic Context-Free Grammar," *IEEE Trans. Instrumentation and Measurement*, vol. 57, no. 8, pp. 1562–1571, Aug. 2008.

[49] Y. Dedeoglu, B. U. Töreyin, U. Güdükbay, and A. E. Çetin, "Silhouette-Based Method for Object Classification and Human Action Recognition in Video." in *ECCV Workshop on HCI*, ser. Lecture Notes in Computer Science, T. S. Huang, N. Sebe, M. S. Lew, V. Pavlovic, M. Kölsch, A. Galata, and B. Kisacanin, Eds., vol. 3979. Springer, 2006, pp. 64–77.

[50] X. Han, X. Chang, L. Quan, X. Xiong, J. Li, Z. Zhang, and Y. Liu, "Feature subset selection by gravitational search algorithm optimization," *Information Sciences*, vol. 281, pp. 128–146, Oct. 2014.

[51] T. Shanableh and K. Assaleh, "User-independent recognition of Arabic sign language for facilitating communication with the deaf community," *Digit. Signal Process.*, vol. 21, no. 4, pp. 535–542, Jul. 2011.

[52] E. Sánchez-Nielsen, L. Antón-Canalís, and M. Hernández-Tejera, "Hand Gesture Recognition for Human-Machine Interaction." in *WSCG*, 2004, pp. 395–402.

[53] S. Conseil, S. Bourennane, and L. Martin, "Comparison of Fourier Descriptors and Hu Moments for HandPosture Recognition," in *European Signal Processing Conference (EUSIPCO)*, 2007.

[54] C. Yu, X. Wang, H. Huang, J. Shen, and K. Wu, "Vision-Based Hand Gesture Recognition Using Combinational Features," in *6th Int. Conf. Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*, Oct. 2010, pp. 543–546.

[55] V. I. Pavlovic, R. Sharma, and T. S. Huang, "Visual interpretation of hand gestures for human-computer interaction: a review," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 677–695, Jul. 1997.

[56] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE Trans. Information Theory*, vol. 8, no. 2, pp. 179–187, Feb. 1962.

[57] S.-K. Hwang and W.-Y. Kim, "A novel approach to the fast computation of Zernike moments," *Pattern Recognition*, vol. 39, no. 11, pp. 2065–2076, Nov. 2006.

[58] J. Triesch and C. von der Malsburg, "A system for person-independent hand posture recognition against complex backgrounds," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 12, pp. 1449–1453, Dec. 2001.

[59] H. Zhou, D. Lin, and T. Huang, "Static Hand Gesture Recognition based on Local Orientation Histogram Feature Distribution Model," in *Conf. Computer Vision and Pattern Recognition Workshop, 2004 (CVPRW '04)*, June 2004, pp. 161–161.

[60] D. Y. Huang, W. C. Hu, and S. H. Chang, "Gabor Filter-based Hand-pose Angle Estimation for Hand Gesture Recognition Under Varying Illumination," *Expert Syst. Appl.*, vol. 38, no. 5, pp. 6031–6042, May 2011.

[61] "THOMAS: Moeslund's Gesture Recognition Database." [Online]. Available: http://www-prima.inrialpes.fr/FGnet/data/12-MoeslundGesture/database.html.

[62] "Michigan State University. (2002). American Sign Language Browser, East Lansing." [Online]. Available: http://commtechlab.msu.edu/sites/aslweb/browser.html.

[63] D. Cvetkovic, E. D. Übeyli, and I. Cosic, "Wavelet transform feature extraction from human PPG, ECG, and EEG signal responses to ELF PEMF exposures: A pilot study," *Digital Signal Processing*, vol. 18, no. 5, pp. 861–874, Sep. 2008.

[64] L. Gupta, T. Sortrakul, A. Charles, and P. Kisatsky, "Robust automatic target recognition using a localized boundary representation," *Pattern Recognition*, vol. 28, no. 10, pp. 1587– 598, Oct. 1995.

[65] S. Ari and G. Saha, "In search of an optimization technique for Artificial Neural Network to classify abnormal heart sounds," *Appl. Soft Comput.*, vol. 9, no. 1, pp. 330–340, Jan. 2009.

[66] C.-N. Fan and F.-Y. Zhang, "Homomorphic Filtering Based Illumination Normalization Method for Face Recognition," *Pattern Recogn. Lett.*, vol. 32, no. 10, pp. 1468–1479, Jul. 2011.

[67] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed. Addison-Wesley Longman Publishing Co., Inc., 2001.

[68] E. Y. Lam, "Combining gray world and retinex theory for automatic white balance in digital photography," in *Proc. 9th Int. Symp. Consumer Electronics, 2005 (ISCE 2005)*, 2005, pp. 134–139.

[69] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," *J. Electronic Imaging*, vol. 13, no. 1, pp. 146–168, Jan. 2004.

[70] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Trans. Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62 –66, Jan. 1979.

[71] J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, Nov. 1986.

[72] T. B. Moeslund, "Canny Edge Detection," Tech. Rep. 09gr820, Aalborg University, March 23 2009. [Online]. Available: http://www.cse.iitd.ernet.in/~pkalra/csl783/canny.pdf

[73] B. Green, "Canny Edge Detection Tutorial," Tech. Rep. Drexel University, Philadelphia, 2002. [Online]. Available: http://dasl.mem.drexel.edu/alumni/bGreen/www.pages.drexel.edu/_weg22/can_tut.html

[74] T. B. Moeslund, *Image and Video Processing*. Computer Vision and Media Technology, Aalborg University, 2008.

[75] D. Chai and A. Bouzerdoum, "A Bayesian approach to skin color classification in YCbCr color space," in *Proc. TENCON 2000.*, vol. 2, 2000, pp. 421–424.

[76] D. Chai and K. Ngan, "Face segmentation using skin-color map in videophone applications," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 9, no. 4, pp. 551–564, Jun 1999.

[77] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed., ser. Springer Series in Statistics. Springer, 2002.

[78] E. Dougherty, *An introduction to morphological image processing*, ser. Tutorial texts in optical engineering. SPIE Optical Engineering Press, 1992.

[79] D. Kelly, J. McDonald, and C. Markham, "A person independent system for recognition of hand postures used in sign language," *Pattern Recogn. Lett.*, vol. 31, no. 11, pp. 1359–1368, Aug. 2010.

[80] S. Haykin, *Neural Networks: A Comprehensive Foundation (3rd Edition).* Prentice-Hall, Inc., 2007.

[81] D. K. Ghosh and S. Ari, "Static Hand Gesture Recognition using Mixture of Features and SVM Classifier," in *Proc. 5th IEEE Int. Conf. Communication Systems and Network Technologies (CSNT-2015)*, Apr 2015.

[82] P. de Chazal, M. O'Dwyer, and R. B. Reilly, "Automatic classification of heartbeats using ECG morphology and heartbeat interval features," *IEEE Trans. Biomedical Engineering*, vol. 51, no. 7, pp. 1196 –1206, Jul. 2004.

[83] E. D. Übeyli, "Statistics over features of ECG signals," *Expert Systems with Applications*, vol. 36, pp. 8758–8767, Jul. 2009.

[84] F. Provost and T. Fawcett, "Analysis and Visualization of Classifier Performance: Comparison under Imprecise Class and Cost Distributions," in *Proc. 3rd Int. Conf. Knowledge Discovery and Data Mining.* AAAI Press, 1997, pp. 43–48.

[85] C. Li and K. M. Kitani, "Pixel-level hand detection in ego-centric videos," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2013, pp. 3570–3577.

[86] H. Badi, S. H. Hussein, and S. A. Kareem, "Feature extraction and ML techniques for static gesture recognition," *Neural Computing and Applications*, vol. 25, no. 3-4, pp. 733–741, Sep. 2014.

[87] A. Atakulreka and D. Sutivong, "Avoiding Local Minima in Feedforward Neural Networks by Simultaneous Learning," in *Proc. 20th Australian Joint Conf. Advances in Artificial Intelligence*, ser. AI'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 100–109.

[88] S. Ari and G. Saha, "In search of an optimization technique for Artificial Neural Network to classify abnormal heart sounds," *Appl. Soft Comput.*, vol. 9, no. 1, pp. 330–340, Jan. 2009.

[89] D. Broomhead and D. Lowe, *Radial Basis Functions, Multi-variable Functional Interpolation and Adaptive Networks*, ser. RSRE memorandum / Royal Signals and Radar Establishment. Royals Signals & Radar Establishment, 1988.

[90] M. Sifuzzaman, M. R. Islam, and M. Z. Ali, "Application of Wavelet Transform and its Advantages Compared to Fourier Transform," *Journal of Physical Sciences*, vol. 13, pp. 121–134, 2009.

[91] P. Goupillaud, A. Grossmann, and J. Morlet, "Cycle-octave and related transforms in seismic signal analysis," *Geoexploration*, vol. 23, no. 1, pp. 85–102, Oct. 1984.

[92] I. Adam and E. Rădoi, *Complex Wavelet Transform: application to denoising.* Editura Politehnică, 2010.

[93] A. Grossmann and J. Morlet, "Decomposition of Hardy functions into square integrable wavelets of constant shape," *SIAM journal on mathematical analysis*, vol. 15, no. 4, pp. 723–736, 1984.

[94] P.-Y. Lin, *An Introduction to Wavelet Transform*, Graduate Institute of Communication Engineering National Taiwan University, Taipei, Taiwan, ROC, 2007.

[95] A. N. Akansu, W. A. Serdijn, and I. W. Selesnick, "Emerging applications of wavelets: A review," *Physical Communication*, vol. 3, no. 1, pp. 1–18, Mar. 2010.

[96] M. Vetterli and C. Herley, "Wavelets and filter banks: theory and design," *IEEE Trans. Signal Processing*, vol. 40, no. 9, pp. 2207–2232, Sep 1992.

[97] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Communications on pure and applied mathematics*, vol. 41, no. 7, pp. 909–996, 1988.

[98] S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, Jul 1989.

[99] G. Saha, S. Senapati, and S. Chakroborty, "An F-Ratio Based Optimization on Noisy Data for Speaker Recognition Application," in *IEEE Indicon Conference*, Dec. 2005, pp. 352 – 355.

[100] S. Ari, D. K. Ghosh, and P. K. Mohanty, "Edge detection using ACO and F ratio," *Signal, Image and Video Processing*, vol. 8, no. 4, pp. 625–634, May 2014.

[101] P.-T. Yap, R. Paramesran, and S.-H. Ong, "Image analysis by Krawtchouk moments," *IEEE Trans. Image Processing*, vol. 12, no. 11, pp. 1367–1377, Nov. 2003.

[102] J. S. Rani and D. Devaraj, "Face recognition using Krawtchouk moment," *Sadhana*, vol. 37, no. 4, pp. 441–460, Aug. 2012.

[103] N. Ahmed, T. Natarajan, and K. Rao, "Discrete Cosine Transform," *IEEE Trans. Computers*, vol. C-23, no. 1, pp. 90–93, Jan. 1974.

[104] A. B. Watson, "Image compression using the discrete cosine transform," *Mathematica journal*, vol. 4, no. 1, p. 81, 1994.

[105] S. A. Khayam, "The discrete cosine transform (DCT): theory and application," Tech. Rep. TR802, Michigan State University, March 2003. [Online]. Available: http://www.lokminglui.com/DCT_TR802.pdf

[106] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Trans. Neural Networks*, vol. 2, no. 2, pp. 302–309, Mar 1991.

[107] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. John Willey & Sons, 1973.

[108] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., 1995.

[109] R. S. Stanković and B. J. Falkowski, "The haar wavelet transform: its status and achievements," *Computers & Electrical Engineering*, vol. 29, no. 1, pp. 25–44, 2003.

[110] C. S. Burrus, R. A. Gopinath, and H. Guo, *Introduction to Wavelets and Wavelet Transforms: A Primer*. Prentice Hall, 1997.

[111] T. B. Dinh, V. B. Dang, D. A. Duong, T. T. Nguyen, and D.-D. Le, "Hand gesture classification using boosted cascade of classifiers," in *Int. Conf. Research, Innovation and Vision for the Future*, Feb. 2006, pp. 139–144.

[112] M. J. Er, S. Wu, J. Lu, and H. L. Toh, "Face recognition with radial basis function (RBF) neural networks," *IEEE Transactions on Neural Networks*, vol. 13, no. 3, pp. 697–710, May 2002.

[113] K. Teknomo, *K-Means Clustering Tutorial*, Ateneo de Manila University, School of Science and Engineering (SOSE), Information Systems & Computer Science, July 2007. [Online]. Available: http://croce.ggf.br/dados/K%20mean%20Clustering1.pdf

[114] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl, "Constrained K-means Clustering with Background Knowledge," in *Proceedings of the Eighteenth International Conference on Machine Learning*, ser. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 577–584.

[115] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, 2002.

[116] J.-R. Tsai, P.-C. Chung, and C.-I. Chang, "A sigmoidal radial basis function neural network for function approximation," in *IEEE Int. Conf. Neural Networks,*, vol. 1, Jun. 1996, pp. 496–501.

[117] M. S. Sainin and R. Alfred, "A genetic based wrapper feature selection approach using nearest neighbour distance matrix," in *3rd Conference on Data Mining and Optimization (DMO)*, June 2011, pp. 237–242.

[118] H. Liu and L. Yu, "Toward integrating features selection algorithms for classification and clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 17, pp. 491–502, 2005.

[119] P. Mitra, C. Murthy, and S. Pal, "Unsupervised feature selection using feature similarity," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 301–312, Mar. 2002.

[120] J. A. S. del Rivero, E. Montañés-Roces, B. de la Roza-Delgado, A. Soldado, O. Luaces, J. R. Quevedo, and A. Bahamonde, "Feature selection for classification of animal feed ingredients from near infrared microscopy spectra." *Inf. Sci.*, vol. 241, pp. 58–69, Aug. 2013.

[121] J. Derrac, C. Cornelis, S. García, and F. Herrera, "Enhancing evolutionary instance selection algorithms by means of fuzzy rough set based feature selection," *Information Sciences*, vol. 186, pp. 73–92, Mar. 2012.

[122] Y. Saeys, I. n. Inza, and P. Larrañaga, "A Review of Feature Selection Techniques in Bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, Sep. 2007.

[123] A. Jain and D. Zongker, "Feature selection: evaluation, application, and small sample performance," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 153–158, Feb. 1997.

[124] M. Hasanuzzaman, S. Saha, and A. Ekbal, "Feature Subset Selection Using Genetic Algorithm for Named Entity Recognition." in *PACLIC*, R. Otoguro, K. Ishikawa, H. Umemoto, K. Yoshimoto, and Y. Harada, Eds. Institute for Digital Enhancement of Cognitive Development, Waseda University, 2010, pp. 153–162.

[125] D. C. Walters and G. B. Sheble, "Genetic algorithm solution of economic dispatch with valve point loading," *IEEE Trans. Power Systems*, vol. 8, no. 3, pp. 1325–1332, Aug. 1993.

[126] R. Rojas, *Neural Networks: A Systematic Introduction.* New York, NY, USA: Springer-Verlag New York, Inc., 1996.

[127] E. Zio, P. Baraldi, and N. Pedroni, "Selecting Features for Nuclear Transients Classification by Means of Genetic Algorithms," *IEEE Trans. Nuclear Science*, vol. 53, no. 3, pp. 1479–1493, Jun. 2006.

[128] H. Huang, Y. Wu, Y. Chan, and C. Lin, "Study on image feature selection: A genetic algorithm approach," in *IET Int. Conf. Frontier Computing. Theory, Technologies and Applications.* Taichung, Taiwan, Aug. 2010, pp. 169–174.

[129] A. K. Sahoo, "Development of Radar Pulse Compression Techniques Using Computational Intelligence Tools," Ph.D. dissertation, National Institute of Technology Rourkela, 2012.

[130] J. H. Holland, *Adaptation in Natural and Artificial Systems.* Ann Arbor: The University of Michigan Press, 1975.

[131] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[132] F. Van Batenburg, A. P. Gultyaev, and C. W. Pleij, "An APL-programmed genetic algorithm for the prediction of RNA secondary structure," *Journal of Theoretical Biology*, vol. 174, no. 3, pp. 269–280, Jun. 1995.

[133] Y. Li, K. C. Ng, D. J. Murray-Smith, G. Gray, and K. C. Sharman, "Genetic algorithm automated approach to the design of sliding mode control systems," *International Journal of Control*, vol. 63, no. 4, pp. 721–739, Feb. 1996.

[134] K. Michail, "Optimised configuration of sensing elements for control and fault tolerance applied to an electro-magnetic suspension system," Ph.D. dissertation, © Konstantinos Michail, 2009.

[135] O. Maimon and D. Braha, "A genetic algorithm approach to scheduling PCBs on a single machine," *International Journal of Production Research*, vol. 36, no. 3, pp. 761–784, 1998.

[136] N. M. Razali and J. Geraghty, "Genetic Algorithm Performance with Different Selection Strategies in Solving TSP," in *Proc. World Congress on Engineering*, vol. 2, 2011.

[137] K. Sastry, D. Goldberg, and G. Kendall, *Genetic Algorithms*. Springer US, 2005, ch. Search Methodologies, pp. 97–125.

[138] D. E. Goldberg and K. Deb, "A comparative analysis of selection schemes used in genetic algorithms," in *Foundations of Genetic Algorithms*. Morgan Kaufmann, 1991, pp. 69–93. [Online]. Available: http://www.cse.unr.edu/~sushil/class/gas/papers/Select.pdf

[139] M. Lozano, F. Herrera, and J. R. Cano, "Replacement strategies to maintain useful diversity in steady-state genetic algorithms," in *Soft Computing: Methodologies and Applications*. Springer, 2005, pp. 85–96.

Publications

- **D. K. Ghosh** and S. Ari, "On an Algorithm for Vision-Based Hand Gesture Recognition," *Signal, Image and Video Processing, Springer*, 2015, vol. 10, no. 4, pp. 655–662, 2016.
- S. Ari, **D. K. Ghosh** and P. K. Mohanty, "Edge detection using ACO and F-ratio," *Signal, Image and Video Processing, Springer*, vol. 8, no. 4, pp. 625–634, 2014.
- **D. K. Ghosh** and S. Ari, "A Static Hand Gesture Recognition Algorithm using K-Mean based Radial Basis Function Neural Network," 8^{th} *International Conference on Information, Communications and Signal Processing (ICICS)*, Singapore, Dec. 13-16, 2011.
- **D. K. Ghosh** and S. Ari, "Static Hand Gesture Recognition using Mixture of Features and SVM Classifier," 5^{th} *IEEE Int. Conf. on Communication Systems and Network Technologies (CSNT-2015)*, Gwalior, India, Apr. 4–6, 2015.
- S. Chatterjee, **D. K. Ghosh**, and S. Ari, "Static Hand Gesture Recognition Based on Fusion of Moments," *in Intelligent Computing, Communication and Devices*, pp. 429–434, Springer India, 2015.
- A. Babu, **D. K. Ghosh**, and S. Ari, "Color Hand Gesture Segmentation for Images with Complex Background," *International Conference on Circuits, Power and Computing Technologies (ICCPCT-2013)*, Kumarcoil, Tamilnadu, Mar. 20–21, 2013.
- **D. K. Ghosh** and S. Ari, "Hand Gesture Recognition using DWT and Sigmoidal Functions based RBF Neural Network," *Submitted to Image and Vision Computing, Elsevier.*

- **D. K. Ghosh** and S. Ari, "Static hand gesture recognition using DWT and an improved RBF neural network," *Submitted to Signal, Image and Video Processing, Springer.*
- **D. K. Ghosh** and S. Ari, "Static Hand Gesture Recognition using DWT and F-ratio based feature set with SVM classifier," *Submitted to International Journal of Computer Applications in Technology, Inderscience.*

◇

Author's Biography

Dipak Kumar Ghosh was born and brought up in Kharigeria, a small village belongs to Anandapur police station, West Bengal and studied in Banerjeedanga high school. After schooling, he started his engineering career and received B.Tech degree in Electronics and communication Engineering from Kalyani Government Engineering College under West Bengal University of Technology in 2005 and subsequently he joined at SAI Services & Technology Pvt. Ltd, Kolkata as System Engineer. After working for one and half years, he received M. Tech degree in Mechatronics from National Institute of Technical Teacher's Training and Research Kolkata under West Bengal University of Technology in 2009. After that, he joined as a lecturer at Mallabhum Institute of Technology. In 2010, he has taken admission for Ph.D. at National Institute of Technology Rourkela. His area of interest is Image processing, Soft computing, Gesture recognition and Pattern classification. During his research, he presented papers in International Conferences. He is also a student Member of IEEE. He can be contacted at: dipakkumar05.ghosh@gmail.com.